



>Three Way Milkshake_

Norme di Progetto

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

Versione	0.4.1
Stato	Redazione
Uso	Interno
Approvazione	Non approvato
Redazione	Nicolò Greggio
Verifica	Non verificato
Destinatari	Three Way Milkshake Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Questo documento contiene tutte le norme di progetto, definite inizialmente o aggiunte in seguito, viene quindi aggiornato per incrementi successivi.



Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
0.5.0	Stesura sezioni § 4.1, 4.2	2021-01-06	Greggio Nicolò	<i>Redattore</i>
0.4.1	Aggiunta sezione § 3.6	2021-01-06	Greggio Nicolò	<i>Redattore</i>
0.4.0	Stesura sezioni § 3.1, 3.2	2021-01-05	Greggio Nicolò	<i>Redattore</i>
0.3.0	Termine stesura sezione § 2	2021-01-04	Greggio Nicolò	<i>Redattore</i>
0.2.2	Stesura sezione § 2.2	2021-01-04	Greggio Nicolò	<i>Redattore</i>
0.2.1	Stesura sezione § 2.1	2021-01-03	Greggio Nicolò	<i>Redattore</i>
0.2.0	Stesura sezione § 1	2021-01-02	Greggio Nicolò	<i>Redattore</i>
0.1.1	Strutturazione nel dettaglio delle sezioni	2020-12-29	Greggio Nicolò	<i>Redattore</i>
0.1.0	Creazione ed inizio strutturazione documento	2020-12-27	Greggio Nicolò	<i>Redattore</i>



Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Termini, abbreviazioni ed altri documenti	5
1.4	Riferimenti	5
2	Processi primari	7
2.1	Fornitura	7
2.1.1	Scopo	7
2.1.2	Aspettative	7
2.1.3	Descrizione	7
2.1.4	Attività	7
2.1.4.1	Inizializzazione	7
2.1.4.2	Preparazione della risposta	7
2.1.4.3	Contratto	7
2.1.4.4	Pianificazione	7
2.1.4.5	Esecuzione e controllo	8
2.1.4.6	Revisione e valutazione	8
2.1.4.7	Consegna e completamento	8
2.2	Sviluppo	8
2.2.1	Scopo	8
2.2.2	Aspettative	8
2.2.3	Descrizione	9
2.2.4	Attività	9
2.2.4.1	Analisi dei requisiti	9
2.2.4.1.1	Descrizione	9
2.2.4.1.2	Nomenclatura casi d'uso	9
2.2.4.1.3	Classificazione dei requisiti	10
2.2.4.2	Progettazione	10
2.2.4.2.1	Descrizione	10
2.2.4.2.2	Obiettivi di qualità	10
2.2.4.2.3	Attività	11
2.2.4.3	Codifica	11
3	Processi di Supporto	12
3.1	Documentazione	12
3.1.1	Scopo	12
3.1.2	Aspettative	12
3.1.3	Descrizione	12
3.1.4	Ciclo di vita dei documenti	12
3.1.5	Struttura dei documenti formali	13
3.1.5.1	Organizzazione in file e cartelle	13
3.1.5.2	Frontespizio	13
3.1.5.3	Registro delle modifiche	14
3.1.5.4	Indice	14
3.1.5.5	Elenchi delle figure e tabelle	14
3.1.5.6	Sezione di introduzione	14



3.1.5.7	Ulteriori sezioni, appendici e glossari	14
3.1.6	Verbalì	14
3.1.6.1	Tracciamento delle decisioni nei verbali interni	15
3.1.7	Glossario e acronimi	15
3.1.7.1	Definizione ed utilizzo delle voci	15
3.1.7.2	Funzionamento	15
3.1.8	Norme tipografiche	16
3.1.8.1	Nomi di file e cartelle	16
3.1.8.2	Stile del testo	16
3.1.8.3	Elenchi	16
3.1.8.4	Data e ora	16
3.1.8.5	Codici di versioni	17
3.1.9	Strumenti	17
3.2	Gestione della configurazione	17
3.2.1	Scopo	17
3.2.2	Aspettative	17
3.2.3	Descrizione	17
3.2.4	Workflow di versionamento	17
3.2.4.1	Pull requests	17
3.2.5	Integrazione con Jira	18
3.2.6	Strumenti	18
3.3	Risoluzione dei problemi	18
3.3.1	Scopo	18
3.3.2	Aspettative	18
3.3.3	Descrizione	18
3.4	Accertamento della qualità	18
3.4.1	Scopo	18
3.4.2	Aspettative	18
3.4.3	Descrizione	19
3.4.4	altre sottosezioni	19
3.5	Verifica	19
3.5.1	Scopo	19
3.5.2	Aspettative	19
3.5.3	Descrizione	19
3.5.4	altre sottosezioni	19
3.5.5	Strumenti	19
3.6	Validazione	19
3.6.1	Scopo	19
3.6.2	Aspettative	19
3.6.3	Descrizione	19
3.6.4	altre sottosezioni	19
3.6.5	Strumenti	19
4	Processi Organizzativi	20
4.1	Gestione dei processi	20
4.1.1	Scopo	20
4.1.2	Aspettative	20
4.1.3	Descrizione	20
4.1.3.1	Jira	20



4.1.3.2	workflow	20
4.1.4	altre sottosezioni	21
4.1.5	Strumenti	21
4.2	I ruoli di progetto	21
4.2.1	Analista	21
4.2.2	Progettista	21
4.2.3	Verificatore	21
4.2.4	Programmatore	22
4.2.5	Responsabile	22
4.2.6	Amministratore	22
4.3	Infrastruttura	22
4.3.1	Scopo	22
4.3.2	Aspettative	22
4.3.3	Descrizione	22
4.3.4	altre sottosezioni	22
4.3.5	Strumenti	22
4.4	Miglioramento	22
4.4.1	Scopo	22
4.4.2	Aspettative	22
4.4.3	Descrizione	23
4.4.4	altre sottosezioni	23
4.4.5	Strumenti	23
4.5	Formazione	23
4.5.1	Scopo	23
4.5.2	Aspettative	23
4.5.3	Descrizione	23
4.5.4	altre sottosezioni	23
4.5.5	Strumenti	23



Elenco delle figure

4.1.1 Diagramma degli stati e delle transizioni del workflow	21
--	----



Elenco delle tabelle



1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di fissare e definire tutte le regole, convenzioni e buone pratiche utili a formare un way of working condiviso alla base da tutti i componenti del gruppo per assicurare una collaborazione efficiente ed efficace. Si discuteranno inoltre i vari strumenti che verranno adottati per facilitare lo sviluppo del progetto e per promuovere un'organizzazione adeguata. Si ritiene inoltre che la definizione ed il mantenimento per incremento di un documento condiviso all'interno del gruppo di lavoro, che definisca e raccolga quanto descritto in maniera formale e centralizzata, possa favorire, in un contesto dove i membri possano variare, l'inserimento di nuovi componenti facilitandone l'ambientamento. Pur non essendo questo il contesto di lavoro attuale, è comunque una buona pratica da sperimentare e consolidare.

1.2 Scopo del prodotto

scopodelprodotto

1.3 Termini, abbreviazioni ed altri documenti

Tutti i termini che necessitano di una spiegazione, per fornire un'adeguata comprensione, o perché possono causare ambiguità nel contesto, sono definiti nel glossario alla fine del documento. Ogni occorrenza di questi collega alla voce corrispondente. Analogamente vale lo stesso discorso per le abbreviazioni e gli acronimi. Le definizioni delle voci nel glossario e nella lista degli acronimi presentano inoltre collegamenti alle pagine dove vengono utilizzati, il che permette una comoda navigazione bidirezionale tra termini e significati corrispondenti. Le voci di glossario saranno seguite da una G pedice mentre gli acronimi da una A (e.g.: voce di glossario_G ; acronimo_A). Inoltre quando si farà riferimento ad un altro documento o al documento stesso, il nome di questo sarà in maiuscoletto (e.g.: ESEMPIO NOME DOCUMENTO).

1.4 Riferimenti

- Standard ISO 12207: https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L03.pdf>
- <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L06.pdf>
- <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/FC2.pdf>
- <https://www.omg.org/spec/UML/2.0/Superstructure/PDF>
- https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20Use%20Case_4x4.pdf
- https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design_4x4.pdf
- <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L09.pdf>
- https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20delle%20Classi_4x4.pdf



- https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20dei%20Package_4x4.pdf
- https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Attivit%c3%a0_4x4.pdf
- https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Sequenza_4x4.pdf
- <https://www.iso.org/iso-8601-date-and-time-format.html>
- <https://docs.github.com/en/free-pro-team@latest/actions>
- <https://www.latex-project.org/>
- <http://www.texstudio.org/>
- <https://www.xmlmath.net/texmaker/>
- <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- <https://github.com/about>
- <https://github.com/about>

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Il processo di fornitura sostanzialmente si occupa della gestione dei rapporti con il cliente. Il suo scopo quindi è quello di determinare strumenti e competenze utili e necessarie alla realizzazione del prodotto e di assicurare la conformità di questo con le richieste del proponente. Si rende necessaria quindi la produzione di documenti che descrivano le intenzioni e le modalità che il gruppo si prefigge di seguire al fine di soddisfare il cliente.

2.1.2 Aspettative

Il confronto diretto e frequente tra fornitore e proponente è senza dubbio utile ad entrambe le parti, affinché ambedue soddisfino i loro obiettivi in tempi desiderabili.

2.1.3 Descrizione

Il processo di fornitura si compone ¹ di 7 attività, definite come segue:

2.1.4 Attività

2.1.4.1 Inizializzazione

Il gruppo dovrà effettuare collettivamente una valutazione di tutti i capitoli proposti e formalizzarla in uno STUDIO DI FATTIBILITÀ, il quale, per ogni capitolo, darà una breve descrizione dello stesso, delle finalità, delle tecnologie, degli aspetti positivi e delle criticità, proponendo poi delle conclusioni che saranno il frutto delle riflessioni interne ed indicando la scelta definitiva dei membri.

2.1.4.2 Preparazione della risposta

Il gruppo preparerà una lettera di presentazione indirizzata al committente ed al proponente del capitolo scelto, per candidarsi alla fornitura del prodotto indicando un sunto del preventivo dei costi.

2.1.4.3 Contratto

2.1.4.4 Pianificazione

Il gruppo dovrà fornire dei documenti che illustrino la gestione del lavoro e mostrino come verranno assicurati qualità e conformità del prodotto. Nello specifico realizzerà:

- UN PIANO DI PROGETTO, contenente ²:
 - **pianificazione macroscopica (a lungo periodo):**
 - scadenze, fissate all'indietro;
 - analisi dei rischi;
 - preventivo dei costi.

¹Standard ISO 12207 § 5.2

²<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/FC2.pdf> § 3



– **pianificazione dettagliata (a breve):**

- attività, fissate in avanti;
- preventivo minuto, alla luce del consuntivo di periodo precedente;
- riscontro dei rischi ed aggiornamento delle misure di mitigazione.

che andrà così strutturato: ³:

- Introduzione (scopo e struttura);
- Organizzazione del progetto;
- Analisi dei Rischi;
- Risorse disponibili (tempo e persone);
- Suddivisione del lavoro (work breakdown);
- Calendario delle attività;
- Meccanismi di controllo e di rendicontazione.

• **UN PIANO DI QUALIFICA**, contenente ⁴:

- obiettivi quantitativi di qualità;
- cruscotto di misurazione;
- analisi degli scostamenti e misure correttive.

2.1.4.5 Esecuzione e controllo

In questa attività il gruppo Three Way Milkshake dovrà implementare ed eseguire i piani delineati al punto 2.1.4.4 e sviluppare il prodotto in accordo con il [processo di sviluppo](#).

2.1.4.6 Revisione e valutazione

Il gruppo dovrà coordinare le revisioni delle attività svolte e gestire la comunicazione con il committente ed il proponente. Si dovrà inoltre aver cura di operare in accordo con quanto scritto negli altri processi.

2.1.4.7 Consegna e completamento

Il fornitore dovrà consegnare il prodotto in accordo con quanto specificato nel contratto. A seguito della consegna del prodotto, il gruppo Three Way Milkshake non si farà carico delle mansioni di supporto ed assistenza.

2.2 Sviluppo

2.2.1 Scopo

Il processo di sviluppo comprende tutte quelle attività che portano alla costruzione del prodotto finale.

2.2.2 Aspettative

Questo processo dev'essere attuato secondo quanto pattuito con proponente e committente, rispettando i loro requisiti e soddisfacendo le loro aspettative. Tutto ciò naturalmente rispettando le norme definite in questo documento

³<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L06.pdf> § 25

⁴<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/FC2.pdf> § 4



2.2.3 Descrizione

Nel processo di sviluppo si individuano ⁵ diverse attività, riassumibili come segue.

2.2.4 Attività

2.2.4.1 Analisi dei requisiti

2.2.4.1.1 Descrizione

Gli analisti devono stabilire, raccogliere e documentare tutti i requisiti stilando un documento che fornirà una base precisa su cui i progettisti si potranno fondare. Dovrà contenere, in accordo con quanto richiesto dal cliente, la raccolta dei **casì d'uso_G**, rappresentati anche tramite diagrammi UML, ed il tracciamento di tutti i requisiti individuati.

2.2.4.1.2 Nomenclatura casi d'uso

Ogni **caso d'uso_G** è univocamente identificato da un codice, secondo lo schema:

UC[caso].[sottoCaso].[sottoSottoCaso]

dove caso, sottoCaso e sottoSottoCaso sono numeri progressivi che partono da 1. Segue poi un breve nome. Inoltre ogni caso dovrà avere le seguenti sezioni (quelle indicate tra parentesi (...) sono opzionali):

- attori primari;
- (attori secondari);
- descrizione;
- (estensioni);
- (inclusioni);
- precondizioni;
- postcondizioni;
textbfscenari principale: uno solo per **caso d'uso_G**, rappresenta il normale flusso degli eventi;
- (scenari alternativi).

Tutti i **casì d'uso_G** non banali verranno accompagnati da diagrammi secondo lo standard **UML_A 2.0** ^{6 7}. Questo è un esempio di **caso d'uso_G** conforme alle regole appena definite:

⁵Standard ISO 12207 § 5.3

⁶<https://www.omg.org/spec/UML/2.0/Superstructure/PDF> § 16

⁷https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20Use%20Case_4x4.pdf



2.2.4.1.3 Classificazione dei requisiti

Per favorire l'organizzazione ed il tracciamento dei requisiti, questi saranno classificati secondo la convenzione che segue:

R[tipo]-[numero progressivo]-[importanza]

dove tipo è uno tra:

- **V** → vincolo;
- **F** → funzionale;
- **Q** → qualità;
- **P** → prestazionale;

mentre importanza può essere:

- **O** → obbligatorio, irrinunciabile per qualcuno degli stakeholder;
- **D** → desiderabile, non strettamente necessario, ma se ne riconosce il valore aggiunto;
- **F** → facoltativo, relativamente utile oppure contrattabile in futuro.

Il numero progressivo parte da 1 ed i sotto requisiti si indicano in maniera analogo ai sotto casi d'uso_G.

2.2.4.2 Progettazione

2.2.4.2.1 Descrizione

I progettisti hanno il compito di tradurre il problema in una possibile soluzione, descritta come architettura, da dividere in parti che possono essere trattate individualmente. Sarà necessario:

- svolgere le attività coerentemente con quanto individuato nell'analisi dei requisiti;
- adottare [design pattern](#)_G opportuni quando si individuano strutture che ne possono trarre beneficio;
- seguire i principi SOLID⁸;
- utilizzare sempre una nomenclatura significativa, parlante e consistente.

2.2.4.2.2 Obiettivi di qualità

Affinché un'architettura si possa definire buona, deve perseguire i seguenti obiettivi di qualità: ⁹

- **sufficienza**: soddisfa tutti i requisiti;
- **comprensibilità**: per tutti gli stakeholder;

⁸https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design_4x4.pdf

⁹<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L09.pdf> § 17..27



- **modularità:** suddivisa in parti chiare e ben distinte;
- **robustezza:** capace di sopportare ingressi diversi da utenti ed ambienti differenti;
- **flessibilità:** permette modifiche a costo contenuto al variare dei requisiti;
- **riusabilità:** le sue parti possono essere impiegate in altre applicazioni;
- **efficienza:** nel tempo (CPU), nello spazio (RAM), nelle comunicazioni (rete);
- **affidabilità:** svolge bene il suo compito quando utilizzata con un'alta probabilità;
- **disponibilità:** la sua manutenzione la rende indisponibile per un tempo ridotto;
- **safety;**
- **security;**
- **semplicità:** ogni parte contiene solo il necessario e niente di superfluo;
- **incapsulazione:** l'interno delle componenti non è visibile all'esterno;
- **coesione:** le parti che stanno insieme hanno gli stessi obiettivi;
- **basso accoppiamento:** parti distinte hanno una dipendenza il più possibile ridotta.

2.2.4.2.3 Attività

La progettazione porterà in seguito alla produzione di una **technology baseline_G** e di una **product baseline_G**. A supporto di queste, ci saranno diversi diagrammi, sempre secondo lo standard **UML_A 2.0**:

- **delle classi¹⁰:** per rappresentare oggetti e relazioni tra questi in un sistema;
- **dei package¹¹:** per rappresentare dipendenze tra classi e package ad un livello di astrazione più alto rispetto a quello dei diagrammi delle classi;
- **delle attività¹²:** per descrivere processi o algoritmi;
- **di sequenza¹³:** la collaborazione di un gruppo di oggetti che devono implementare collettivamente un comportamento.

Si dovranno indicare le tecnologie che si andranno ad utilizzare, descrivendo il loro impiego e le motivazioni che hanno portato alla loro scelta.

2.2.4.3 Codifica

Qui verranno definite le norme che i programmatori dovranno seguire, di modo da:

- perseguire: leggibilità, uniformità e consistenza;
- agevolare: verifica, validazione e manutenzione.

¹⁰https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20delle%20Classi_4x4.pdf

¹¹https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20dei%20Package_4x4.pdf

¹²https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Attivit%c3%a0_4x4.pdf

¹³https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20di%20Sequenza_4x4.pdf

3 Processi di Supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo del processo di documentazione è regolamentare la creazione e la gestione dei documenti e fissare le modalità di stesura ed approvazione degli stessi.

3.1.2 Aspettative

Avere un approccio condiviso ed uniforme per la stesura e l'aggiornamento dei documenti all'interno del gruppo di lavoro è fondamentale per rendere la documentazione uno strumento costruttivo e di supporto, e non un mera formalità aggiuntiva. Inoltre fornire un aspetto uniforme attraverso tutti i documenti facilita qualunque lettore.

3.1.3 Descrizione

Il gruppo Three Way Milkshakesi doterà di due categorie di documentazione:

- **formale:** documenti interni o esterni che rispetteranno strettamente i vincoli descritti in seguito e che saranno interamente pubblici, realizzati in \LaTeX aderendo ad un template condiviso;
- **informale:** documenti interni che potranno svolgere diverse funzioni, tra cui:
 - raccolta appunti e ordini del giorno per riunioni;
 - raccolta argomenti delle discussioni delle riunioni, per tracciare l'evoluzione delle stesse e favorire la stesura dei verbali in seguito;
 - creazione di [wiki_G](#) per condivisione di materiale utile riguardo l'uso di tecnologie o strumenti a supporto di qualsiasi attività.

Questi documenti saranno realizzati sfruttando [Confluence](#) per garantire semplicità, accentramento e condivisione real-time.

3.1.4 Ciclo di vita dei documenti

Ogni documento formale si redige ed incrementa tramite queste attività:

- **stesura:** la scrittura del documento in sé, riguarda sia la creazione di nuove parti che l'aggiornamento di queste. Uno o più redattori si occupano di ciò;
- **verifica:** eseguita da uno o più verificatori, necessariamente diversi dai redattori, consiste nel controllo della correttezza sintattica, semantica, grammaticale ed ortografica e della conformità del documento rispetto alle suo scopo. Nel caso in cui si rendano necessarie modifiche sostanziali, i verificatori notificheranno il responsabile che provvederà a riportare il documento in stesura e solleciterà i redattori affinché apportino le correzioni richieste;
- **approvazione:** quando i verificatori riporteranno la completa correttezza ed aderenza ai requisiti del documento, il responsabile provvederà all'approvazione finale ed al rilascio di una nuova versione dello stesso.

Adottando un approccio incrementale, queste attività possono ripetersi.



3.1.5 Struttura dei documenti formali

3.1.5.1 Organizzazione in file e cartelle

Ogni documento, ha una sua cartella dedicata, all'interno della quale ci devono essere:

- **file principale del documento:** (nome_documento.tex) che contiene l'impostazione della struttura, le dichiarazioni per importare i package \LaTeX aggiuntivi e le dichiarazioni di inclusione delle sezioni e del glossario;
- **cartella config:** contenente file di configurazione relativi al documento, che consentono l'impostazione del frontespizio e del registro delle modifiche;
- **cartella res:** per contenere le risorse del documento, a sua volta questa contiene:
 - **cartella images:** per le eventuali immagini;
 - **cartella sections:** che contiene tutte le sezioni, ognuna in un file diverso, i quali seguono la convenzione di nomenclatura preceduta da un identificativo numerico progressivo ad indicare la posizione della stessa nel documento (e.g.: 01_introduzione.tex, 02_sezione1.tex)
- **file glossario.txt:** dove definire tutti gli acronimi e le voci di glossario, che verranno poi processate dall'automazione.

3.1.5.2 Frontespizio

Fornisce delle informazioni generali e di introduzione al documento, ed è così composto:

- logo esteso del gruppo;
- nome del documento;
- nome del gruppo e del progetto relativo al capitolato scelto;
- indirizzo email di riferimento del gruppo;
- versione del documento, secondo la [convenzione](#);
- stato, in accordo con quanto descritto nel [ciclo di vita dei documenti](#);
- elenco dei redattori;
- elenco dei verificatori;
- nome del responsabile che ha effettuato l'ultima approvazione
- destinatari del documento
- breve descrizione



3.1.5.3 Registro delle modifiche

Raccoglie in forma tabellare tutte le modifiche e conseguentemente la storia del documento, così strutturate:

- versione del documento relativa alla modifica
- breve descrizione delle attività svolte
- data della modifica, secondo lo standard [ISO 8601](#)
- cognome e nome di chi ha apportato la modifica
- ruolo del modificatore rispetto al ciclo di vita del documento, può quindi essere: redattore, verificatore, responsabile.

3.1.5.4 Indice

Riassume i contenuti del documento raccolti per sezioni ed intestazioni, indicando la pagina di inizio e collegando alla stessa

3.1.5.5 Elenchi delle figure e tabelle

Nel caso in cui un documento presenti una o più figure o tabelle, queste sezioni le indicheranno raccogliendole rispettivamente ed indicando la pagina di apparizione.

3.1.5.6 Sezione di introduzione

La prima sezione di contenuto di ogni documento formale è l'introduzione che si articola in:

- una sezione che descrive lo scopo del documento;
- due sezioni condivise fra tutti i documenti, non verbali, che illustrano rispettivamente scopo del prodotto del capitolato scelto e funzionamento di acronimi, voci di glossario e riferimenti ad altri documenti;
- una sezione che contiene tutti i riferimenti, di natura normativa o informativa.

3.1.5.7 Ulteriori sezioni, appendici e glossari

Come indicato precedentemente ogni sezione di primo livello deve avere un suo file dedicato e si svilupperà in seguito all'introduzione. Al termine di tutte le sezioni vi può essere un'appendice mentre le ultime pagine elencano la lista degli acronimi e i termini di glossario utilizzati nel documento, con riferimenti alle pagine dove questi appaiono.

3.1.6 Verbali

I verbali possono essere interni, se relativi ad una riunione dei soli membri del gruppo, o esterni, se relativi ad un qualche tipo di incontro con persone esterne al gruppo. Come gli altri documenti formali, hanno un frontespizio ed un registro delle modifiche, poi si articolano in questo modo:

- informazioni generali
 - dettagli sull'incontro
 - luogo, data¹⁴, orari di inizio e fine e partecipanti
 - ordine del giorno;

¹⁴Standard ISO 8601



- verbale della riunione
 - contiene tutte le sottosezioni necessarie a descrivere lo svolgimento dell'incontro;
- conclusioni
 - nel caso di verbale interno, una tabella di tracciamento delle decisioni;
 - nel caso di verbale esterno, un paragrafo riassuntivo.

Ogni verbale avrà come nome "verbale_[tipo]_[num]" dove tipo può essere interno o esterno, e num è il progressivo rispetto al tipo.

3.1.6.1 Tracciamento delle decisioni nei verbali interni

Ogni riga della tabella di tracciamento delle decisioni si compone di:

- **codice:** identificativo univoco della decisione, così formato: VI_numRiunione.numDecisione (e.g.: VI_2.3 indica la terza decisione presa durante la riunione interna 2);
- **decisione:** breve riassunto che indichi in maniera chiara la decisione presa.

3.1.7 Glossario e acronimi

Ogni documento ha una sua sezione dedicata al glossario e agli acronimi a fine documento, dove sono indicate anche le pagine di apparizione delle rispettive voci.

3.1.7.1 Definizione ed utilizzo delle voci

Gli acronimi ed i termini di glossario vanno definiti nel glossario.txt, istruzioni precise su come utilizzare correttamente questo documento si trovano nella WIKI HOW-TO GLOSSARIO. Una volta che una voce è stata aggiunta, la si può usare liberamente in qualsiasi sezione tramite il suo nome, singolare o plurale nel caso di termine nel glossario, o abbreviazione nel caso di acronimo

3.1.7.2 Funzionamento

Ad ogni azione di [push_G](#) su un branch feature, un'automazione [github action](#) si occupa di eseguire uno script [bash_G](#), `glossary_builder.sh` il quale:

- controlla tutti i documenti che hanno un glossario.txt;
- ne genera un glossario.tex;
- sostituisce tutte le occorrenze delle voci di glossario e acronimi in ogni sezione con i rispettivi comandi \LaTeX
- se ci sono stati cambiamenti, ricompila il glossario e ricompila il documento per ottenere un pdf aggiornato;
- se ci sono stati dei documenti aggiornati, effettua [commit_G](#) e [push_G](#).

Questo permette di avere sempre i file pdf dei documenti aggiornati e con le voci di glossario correttamente compilate e collegate, in maniera automatica e rapida. Si consiglia quindi ai membri di effettuare sempre una [pull_G](#) prima di apportare nuove modifiche ai documenti se si ha precedentemente eseguito una [push_G](#), così da avere il documento aggiornando ed evitare problemi di conflitti e nel [VCS_A](#).



3.1.8 Norme tipografiche

3.1.8.1 Nomi di file e cartelle

Ogni file e cartella dovrà avere un nome:

- che sia il più possibile conciso ed esplicativo;
- composto di sole lettere minuscole, numeri, _ e '-', ad eccezione del '.' per separare nome da estensione
- che abbia alla fine un'estensione coerente col suo tipo
- non contenere spazi o caratteri diversi da quelli indicati sopra, parole diverse si separano con '-'

3.1.8.2 Stile del testo

Ai seguenti stili si attribuisce una specifica funzione semantica:

- **corsivo:** per denotare termini tecnici appartenenti ad una particolare tecnologia che si sta trattando;
- **grassetto:** per evidenziare termini rilevanti o dei quali viene dato un significato esteso immediatamente in seguito, come questi in un elenco puntato;
- **maiuscoletto:** per indicare altri documenti (e.g.: PIANO DI PROGETTO);
- **pedice:** G e A, per indicare rispettivamente una voce di glossario o un acronimo.

3.1.8.3 Elenchi

Ogni elemento deve terminare con un punto e virgola, eccetto quella finale che va seguito da un punto, quindi ogni prima parola deve avere iniziale minuscola a meno che non indichi qualcosa per la quale la capitalizzazione è importante. Gli elenchi puntati non numerati avranno come simbolo di primo livello un puntino pieno, come secondo un trattino, come terzo un asterisco. Gli elenchi numerati come primo livello numeri, come secondo lettere minuscole, come terzo numeri romani rappresentati con lettere minuscole

- | | |
|-------------------------------|----------------------------|
| • elenco puntato non numerato | 1. elenco puntato numerato |
| • secondo elemento | 2. secondo elemento |
| – secondo livello | (a) secondo livello |
| * terzo livello | i. terzo livello |

(Esempi elenchi puntati numerati e non corretti)

3.1.8.4 Data e ora

In ogni documento si adotta lo [Standard ISO 8601](#), quindi aaaa-mm-gg (e.g.: 5 Gennaio 2021 = 2021-01-05) e hh:mm. Se si scrive il mese in caratteri questo con l'iniziale maiuscola e vale lo stesso per i giorni della settimana.



3.1.8.5 Codici di versioni

Si adotta la convenzione $x.y.z$ dove x , y e z sono numeri progressivi che partono da 0, x non si azzera mai, y si azzera ad ogni incremento di x e z si azzera ad ogni incremento di y o x . Gli incrementi avvengono:

- **per x :** quando avviene un accertamento da parte del responsabile di modifiche sostanziali
- **per y :** quando viene apportato un insieme discreto di modifiche;
- **per z :** per ogni modifica o verifica minore.

3.1.9 Strumenti

Come già citato, per la scrittura dei documenti formali si usa $\text{L}^{\text{T}}\text{E}^{\text{X}}$ ¹⁵ e come editor non c'è una preferenza assoluta, ma si consigliano [TexStudio](#) o [TexMaker](#). Per quanto riguarda i documenti informali invece, [Confluence](#) può essere usato in qualunque browser.

3.2 Gestione della configurazione

3.2.1 Scopo

Lo scopo del processo di gestione della configurazione è quello di applicare procedure amministrative e tecniche al fine di controllare e registrare modifiche e rilasci ed assicurare la correttezza, consistenza, completezza ed archiviazione di ogni elemento.

3.2.2 Aspettative

Avere uno spazio di lavoro versionato, condiviso ed altamente orientato alla collaborazione è fondamentale in gruppo di lavoro con molteplici componenti. La storicizzazione, tracciabilità e reversibilità di ogni cambiamento permettono di concentrarsi sulla produzione di nuovo valore invece che sulle problematiche di versionamento, conflitti e condivisione.

3.2.3 Descrizione

Il gruppo Three Way Milkshake ha un'organizzazione su [github](#)¹⁶ all'interno della quale gestisce tutte le [repo_A](#) di cui necessita.

3.2.4 Workflow di versionamento

Il gruppo adotta il workflow [gitflow](#) all'interno di ciascuno [repo_A](#). Per semplificare le operazioni si può usare il pacchetto di estensioni [git-flow](#). Maggiori informazioni ed istruzioni per l'uso di [git_G](#) e GitHub si trovano nella [wiki_G](#) GIT, GITHUB & JIRA

3.2.4.1 Pull requests

Per l'implementazione delle feature dai branch di lavoro, bisogna aprire una [pull_G](#) request su develop. A questo punto il responsabile o l'amministratore provvederanno a fare i controlli necessari ed effettueranno il merge¹⁷ tramite *squash*.

¹⁵<https://www.latex-project.org/>

¹⁶Three Way Milkshake su Github: <https://github.com/Three-Way-Milkshake>

¹⁷Il merge in [git_G](#): <https://www.atlassian.com/git/tutorials/using-branches/git-merge>



3.2.5 Integrazione con Jira

GitHub è stato configurato in maniera che ad ogni `commitG` si scatenino delle azioni che coinvolgono Jira. Usando una specifica sintassi^{18 19} è possibile collegare ogni `commitG` con una task o sottotask di Jira, aggiungendo direttamente su questa commenti, tracciamento del tempo di lavoro ed apportando modifiche allo stato di progresso.

3.2.6 Strumenti

Come già detto, le `repoA` sono tenute su GitHub, per quanto riguarda la gestione locale, ogni membro è libero di adottare lo strumento che preferisce.

3.3 Risoluzione dei problemi

3.3.1 Scopo

Questo processo ha lo scopo di definire un insieme di procedure atte alla risoluzione dei problemi.

3.3.2 Aspettative

Identificazione, segnalazione e risoluzione dei problemi devono avvenire in maniera standardizzata all'interno del gruppo così che tutto il processo possa essere svolto nei tempi più brevi possibili.

3.3.3 Descrizione

Quando un verificatore riscontra dei problemi, tramite il meccanismo degli *smart commits*, riporta la task o sottotask relativa al problema individuato nello stato *in progress* ed aggiunge un commento che indica i problemi riscontrati. Nel caso sia necessario aggiungere più informazioni, il verificatore si può recare nella board ed aggiungere ulteriori commenti. Il verificatore viene quindi sollevato automaticamente dal compito, il responsabile controlla il problema segnalato, se necessario imposta una priorità diversa da *medium*, aggiunge eventuali commenti o descrizioni e riassegna la task ad un altro membro. Il responsabile può sfruttare le sezioni *history* e *comments* di ogni task per controllare chi ha lavorato in passato a quel compito di modo da facilitare la scelta di assegnazione.

3.4 Accertamento della qualità

3.4.1 Scopo

Lo scopo del processo di ...

3.4.2 Aspettative

...

¹⁸<https://support.atlassian.com/bitbucket-cloud/docs/use-smart-commits/>

¹⁹wiki GIT, GITHUB & JIRA



3.4.3 Descrizione

...

3.4.4 altre sottosezioni

...

3.5 Verifica

3.5.1 Scopo

Lo scopo del processo di ...

3.5.2 Aspettative

...

3.5.3 Descrizione

...

3.5.4 altre sottosezioni

...

3.5.5 Strumenti

...

3.6 Validazione

3.6.1 Scopo

Lo scopo del processo di ...

3.6.2 Aspettative

...

3.6.3 Descrizione

...

3.6.4 altre sottosezioni

...

3.6.5 Strumenti

...

4 Processi Organizzativi

4.1 Gestione dei processi

4.1.1 Scopo

Qui lo scopo è la gestione delle attività e dei compiti per ogni altro processo.

4.1.2 Aspettative

L'organizzazione e la gestione dei compiti all'interno del gruppo di lavoro deve avvenire in modo [sistematico_G](#), [disciplinato_G](#) e [quantificabile_G](#) così da favorire uno svolgimento dei lavori ordinato, che rispetti i tempi e che non sia vulnerabile a variazioni e problemi che si possono certamente riscontrare.

4.1.3 Descrizione

Il responsabile e l'amministratore si sono occupati della creazione e della configurazione di tutti gli ambienti necessari:

- account gmail;
- organizzazione GitHub;
- calendario condiviso su Google Calendar;
- workspace [Slack](#);
- workspace Confluence con spazio dedicato ai verbali e spazio dedicato alle wiki;
- progetto Jira con board condivisa per l'organizzazione, gestione e tracciamento dei compiti.

4.1.3.1 Jira

Come spiegato nella sezione [Integrazione con Jira](#), è stato configurato un processo automatico di collegamento fra [commit_G](#) su GitHub e task in Jira. Inoltre ad ogni azione, manuale o automatica, su Jira, viene generata una notifica nel canale dedicato di Slack.

4.1.3.2 workflow

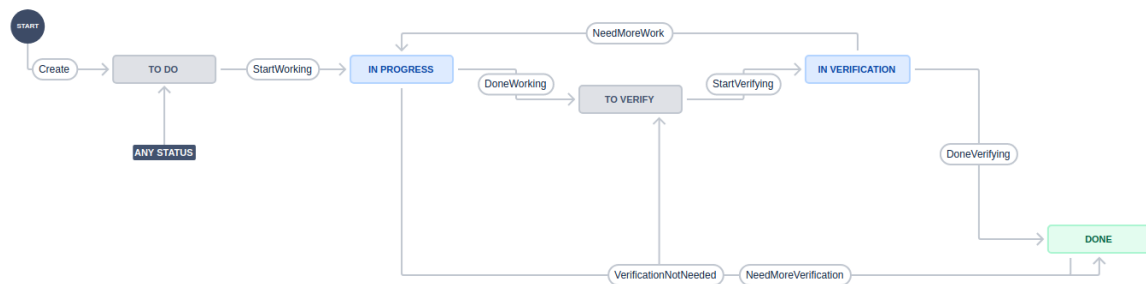


Figura 4.1.1: Diagramma degli stati e delle transizioni del workflow

Come si può vedere dall'immagine, il workflow di task e sottotask è stato configurato di modo che ci siano delle precise transizioni da uno stato all'altro e che solo quelle siano permesse. I nomi delle transizioni sulle frecce corrispondono ai comandi da usare nella sintassi degli smart [commits_G](#). Maggiori dettagli si trovano nella [wiki_G](#) GIT, GITHUB & JIRA.

4.1.4 altre sottosezioni

...

4.1.5 Strumenti

...

4.2 I ruoli di progetto

La suddivisione in ruoli è necessaria per favorire la parallelizzazione e distribuzione del lavoro e delle responsabilità.

4.2.1 Analista

Hai il compito specifico di comprendere il problema. Ha un ruolo attivo fin quando la comprensione non è adeguata. Adottando un modello incrementale, rimane più a lungo ed evolve. È auspicabile che ci siano più analisti per portare punti di vista diversi e sommare le conoscenze. Redige STUDIO DI FATTIBILITÀ ed ANALISI DEI REQUISITI.

4.2.2 Progettista

Traduce il problema in una possibile soluzione, descritta come architettura, divisa in parti per agevolare lo sviluppo individuale di queste. Dati i vincoli ha il compito di trovare una buona soluzione. Redige le specifiche tecniche e le definizioni del prodotto.

4.2.3 Verificatore

Agisce su ogni attività ed attua una verifica oggettiva, segnalando eventuali problemi secondo il textprocesso rispettivo.



4.2.4 Programmatore

Implementa in codice ciò che i progettisti hanno definito come design. Un'adeguata divisione dei compiti tra diversi programmatori consente un alto parallelismo, ideale per avanzare rapidamente. Deve svolgere compiti piccoli, che siano facilmente verificabili.

4.2.5 Responsabile

Gestisce il controllo sul progetto e tramite un cruscotto di controllo aggiorna, revisiona ed adatta lo svolgimento dei lavori. Elabora piani e scadenze, assegna compiti, approva documenti, gestisce i problemi, redige l'ORGANIGRAMMA ed il PIANO DI PROGETTO ed approva l'offerta prima di sottoporla al committente.

4.2.6 Amministratore

Ha in carico la definizione, modifica ed agevolazione del way of working. È un percorso in logica JiT_A procede per incrementi. È responsabile dell'efficacia e dell'efficienza nell'ambiente di lavoro, gestisce la documentazione, organizza il glossario, collabora alla redazione del PIANO DI PROGETTO e redige le NORME DI PROGETTO.

4.3 Infrastruttura

4.3.1 Scopo

Lo scopo del processo di ...

4.3.2 Aspettative

...

4.3.3 Descrizione

...

4.3.4 altre sottosezioni

...

4.3.5 Strumenti

...

4.4 Miglioramento

4.4.1 Scopo

Lo scopo del processo di ...

4.4.2 Aspettative

...



4.4.3 Descrizione

...

4.4.4 altre sottosezioni

...

4.4.5 Strumenti

...

4.5 Formazione

4.5.1 Scopo

Lo scopo del processo di ...

4.5.2 Aspettative

...

4.5.3 Descrizione

...

4.5.4 altre sottosezioni

...

4.5.5 Strumenti

...



Lista degli Acronimi

JiT Just in Time. [22](#)

repo repository. [17](#), [18](#)

UML Unified Modeling Language. [9](#), [11](#)

VCS Version Control System. [15](#)

Glossario dei Termini

bash È una shell Unix ed un linguaggio interpretato scritto da Brian Fox per il progetto GNU come sostituto del software gratuito per la shell Bourne. [15](#)

caso d'uso Un caso d'uso è un insieme di scenari (sequenze di azioni) che hanno in comune uno scopo finale (obiettivo) per un utente (attore). [9](#), [10](#)

commit Comando di [git](#) che permette di salvare e versionare le modifiche attuate ai file in locale. [15](#), [18](#), [20](#), [21](#)

design pattern Una soluzione progettuale generale ad un problema ricorrente. [10](#)

disciplinato Che segue le regole che si è dato. [20](#)

git Sistema di controllo del versionamento distribuito. [17](#)

product baseline G. [11](#)

pull Comando di [git](#) che permette di aggiornare il proprio [repo](#) locale con i cambiamenti remoti. [15](#), [17](#)

push Comando di [git](#) che permette di caricare le proprie modifiche locali sul server remoto condiviso del repository. [15](#)

quantificabile Che permette di misurare l'efficienza e l'efficacia del suo agire. [20](#)

sistematico Modo di lavorare metodico e rigoroso, che conosce, usa ed evolve le best practice di dominio. [20](#)

technology baseline Motiva le tecnologie, i framework, e le librerie selezionate per la realizzazione del prodotto. Ne dimostra adeguatezza e fattibilità, tramite un proof of concept coerente con gli obiettivi. [11](#)

wiki Termine di origine hawaiana che significa veloce, con cui si identifica un tipo di sito internet che permette la creazione e la modifica di pagine multimediali attraverso un'interfaccia semplice. [12](#), [17](#), [21](#)