



Norme di Progetto

Gruppo VarTmp7 - Progetto Stalker vartmp7@gmail.com

Informazioni sul documento

Versione	1.0.0
Approvatore	Xiaowei Wen
Redattori	Giuseppe Zatta Riccardo Tassetto
Verificatori	Stefano Cavaliere Lorenzo Taschin
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin VarTmp7

Descrizione

In questo documento sono descritte le regole, le convenzioni e gli strumenti adottati durante lo sviluppo del progetto Stalker dal gruppo VarTmp7.

Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0	2019-12-30	Documento approvato e pronto per la consegna	Xiaowei Wen	<i>Responsabile</i>
0.3.0	2019-12-29	Terza revisione generale con esito positivo	Stefano Cavaliere	<i>Verificatore</i>
0.2.2	2019-12-28	Correzioni varie sezione 4	Giuseppe Zatta	<i>Redattore</i>
0.2.1	2019-12-28	Correzioni varie sezione 3	Riccardo Tassetto	<i>Redattore</i>
0.2.0	2019-12-27	Seconda revisione generale	Lorenzo Taschin	<i>Verificatore</i>
0.1.4	2019-12-26	Termine stesura sezione 3 e 4	Riccardo Tassetto	<i>Redattore</i>
0.1.3	2019-12-25	Continuazione stesura Processi organizzativi	Riccardo Tassetto	<i>Redattore</i>
0.1.2	2019-12-22	Correzione sezione 3 e 4 dopo verifica	Riccardo Tassetto	<i>Redattore</i>
0.1.1	2019-12-22	Correzione sezione 1 e 2 dopo verifica	Giuseppe Zatta	<i>Redattore</i>
0.1.0	2019-12-22	Prima revisione generale	Stefano Cavaliere, Lorenzo Taschin	<i>Verificatore</i>
0.0.6	2019-12-19	Terminata stesura sezione 4 e modifica Processi di supporto	Riccardo Tassetto	<i>Redattore</i>
0.0.5	2019-12-16	Terminata prima stesura sezione 3	Riccardo Tassetto	<i>Redattore</i>
0.0.4	2019-12-13	Inizio stesura sezioni 3 e 4	Riccardo Tassetto	<i>Redattore</i>
0.0.3	2019-12-12	Scrittura Sviluppo in Processi Primari	Giuseppe Zatta	<i>Redattore</i>
0.0.2	2019-12-02	Scrittura Introduzione e Processi Primati	Giuseppe Zatta	<i>Redattore</i>
0.0.1	2019-11-24	Creazione scheletro del documento e sezione introduzione	Claudia Zattara	<i>Redattore</i>

Indice

1	Introduzione	6
1.1	Scopo del Documento	6
1.2	Glossario	6
1.3	Maturità del Documento	6
1.4	Riferimenti	6
1.4.1	Normativi	6
2	Processi Primari	7
2.1	Fornitura	7
2.1.1	Attività	7
2.1.1.1	Studio di Fattibilità	7
2.1.1.2	Piano di Progetto	7
2.1.1.3	Piano di Qualifica	8
2.1.2	Strumenti	8
2.1.2.1	Microsoft Project	8
2.1.2.1.1	Descrizione Generale	8
2.1.2.1.2	Applicazione al Progetto	8
2.1.2.2	Microsoft Excel	9
2.1.2.2.1	Descrizione Generale	9
2.1.2.2.2	Applicazione al Progetto	9
2.2	Sviluppo	10
2.2.1	Attività	10
2.2.1.1	Analisi Dei Requisiti	10
2.2.1.1.1	Casi d'uso	10
2.2.1.1.2	Requisiti	11
2.2.1.2	Progettazione	11
2.2.1.3	Codifica	11
2.2.1.3.1	Stili di codifica	11
3	Processi di supporto	13
3.1	Documentazione	13
3.1.1	Scopo	13
3.1.2	Ciclo di vita del documento	13
3.1.3	Template	13
3.1.4	Struttura dei documenti	13
3.1.4.1	Prima pagina	13
3.1.4.2	Registro delle modifiche	14
3.1.4.3	Indice	14
3.1.4.4	Elenco immagini	14
3.1.4.5	Elenco tabelle	14
3.1.4.6	Contenuto principale	14
3.1.5	Classificazione dei documenti	15
3.1.5.1	Documenti ad uso interno	15
3.1.5.2	Documenti ad uso esterno	15
3.1.5.3	Verbali	15
3.1.6	Norme tipografiche	16
3.1.6.1	Nomi dei file	16

3.1.6.2	Separatore migliaia	16
3.1.6.3	Elenchi puntati	16
3.1.6.4	Glossario	17
3.1.6.5	Sigle	17
3.1.7	Componenti grafiche	17
3.1.7.1	Tabelle	17
3.1.7.2	Immagini	18
3.1.7.3	Diagrammi UML	18
3.1.8	Strumenti	18
3.1.8.1	L ^A T _E X	18
3.1.8.2	Astah UML	18
3.2	Gestione della configurazione	19
3.2.1	Versionamento	19
3.2.1.1	Codifica della versione del documento	19
3.2.2	Repository	19
3.2.2.1	Tecnologie e strumenti	19
3.2.2.2	Struttura del repository	19
3.2.2.3	Utilizzo di GitKraken	20
3.2.2.4	Gestione delle modifiche	20
3.3	Gestione della qualità	21
3.3.1	Strumenti	21
3.4	Verifica	22
3.4.1	Attività	22
3.4.1.1	Analisi	22
3.4.1.1.1	Analisi Statica	22
3.4.1.1.2	Analisi Dinamica	22
3.4.1.2	Test	22
3.4.1.2.1	Test di unità	22
3.4.1.2.2	Test di integrazione	23
3.4.1.2.3	Test di sistema	23
3.4.1.2.4	Test di regressione	23
3.4.1.2.5	Test di accettazione	23
3.4.2	Strumenti	23
3.4.2.1	Verifica ortografica	23
3.4.2.2	Verifica codice	23
3.4.2.3	Indice di Gulpease	23
3.5	Validazione	24
3.5.1	Attività	24
4	Processi organizzativi	25
4.1	Gestione organizzativa	25
4.1.1	Ruoli di progetto	25
4.1.1.1	Responsabile di progetto	25
4.1.1.2	Amministratore di progetto	25
4.1.1.3	Analista	25
4.1.1.4	Progettista	25
4.1.1.5	Verificatore	25
4.1.1.6	Programmatore	25
4.1.2	Procedure	25

4.1.2.1	Gestione delle comunicazioni	25
4.1.2.2	Gestione degli incontri	26
4.1.2.2.1	Incontri interni	26
4.1.2.2.2	Incontri esterni	26
4.1.3	Formazione	26
4.1.3.1	Conoscenze pregresse	26
4.1.4	Strumenti di coordinamento	26
4.1.4.1	Glo Boards	26

Elenco delle tabelle

2	Esempio corretto di tabella.	18
3	Lista di controllo per la verifica	22

1 Introduzione

1.1 Scopo del Documento

Lo scopo di questo documento è di definire le best practice e il way of working che ogni membro del gruppo *VarTmp7* è tenuto ad adottare durante tutto il ciclo di vita del progetto *Stalker* al fine di garantire quanto più possibile l'uniformità del materiale prodotto.

1.2 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio impiegato nei documenti viene fornito il *Glossario_v.1.0.0*, contenente la definizione dei termini in corsivo marcati con una G a pedice.

1.3 Maturità del Documento

Questo documento sarà soggetto a futuri incrementi per cui non si può definire completo alla attuale versione del documento. Questa decisione è presa al fine di trattare preventivamente e con maggior cura i problemi maggiormente impellenti e ricorrenti. È possibile trovare la pianificazione degli incrementi nel capitolo 4 del *PianoDiProgetto_v.1.0.0*.

1.4 Riferimenti

1.4.1 Normativi

- ISO-IEC 12207:1997:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;
- capitolato d'appalto C5:
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C5.pdf>;
- ISO 31-0: <https://bsol.bsigroup.com/Bibliographic/BibliographicInfoData/000000000030150465>.

2 Processi Primari

2.1 Fornitura

Il processo di fornitura è un processo essenziale nello sviluppo software, ed infatti è uno dei processi primari. Il suo principale scopo è quello di fornire al cliente il prodotto da lui richiesto, ovvero che soddisfa le sue richieste (requisiti). Per fare ciò è necessario determinare le *procedure_G* e le risorse necessarie al corretto svolgimento del progetto. A tale fine è necessario:

- comprendere le richieste del proponente e da ciò determinarne la fattibilità;
- redigere un *PianoDiProgetto_v.1.0.0* che ci aiuti a portare a termine tutte le richieste;
- mantenere una discussione proficua con il proponente in merito ai seguenti temi di discussione:
 - determinare gli aspetti prioritari;
 - determinare vincoli su requisiti e processi;
 - stimare i costi (temporali) del lavoro;
 - accordarsi sulla qualifica del prodotto.

2.1.1 Attività

2.1.1.1 Studio di Fattibilità

L'obiettivo di questa attività è analizzare ogni *capitolato_G* proposto al fine di discutere le varie possibilità che poi sfoceranno nello *StudioDiFattibilità_v.1.0.0* e di conseguenza nella scelta del capitolato stesso. In tal documento, per ogni capitolato proposto verranno indicati:

- informazioni sul capitolato;
- descrizione ed obiettivo finale;
- dominio tecnologico;
- considerazioni del gruppo.

Queste ultime, a loro volta, saranno suddivise in:

- aspetti positivi;
- criticità e fattori di rischio;
- conclusioni.

2.1.1.2 Piano di Progetto

Il *PianoDiProgetto_v.1.0.0*, redatto dal *Responsabile* affiancato dagli *Amministratori* ha come obiettivo la pianificazione delle *attività_G* che guideranno il processo durante le tutte le *fasi_G* del progetto. A questo fine vengono dunque organizzate le risorse disponibili, che verranno quindi distribuite lungo i vari periodi di tempo previsti dal *PianoDiProgetto_v.1.0.0*. In particolare questo documento dovrà affrontare le seguenti tematiche:

- **Analisi dei Rischi:** si analizzano i rischi nei quali si potrebbe incorrere durante lo sviluppo del progetto. Questi rischi vengono poi classificati per gravità e per la probabilità di incorrere in essi;

- **Modello di ciclo di vita:** si definisce il *modello di ciclo di vita_G* scelto per questo progetto, ivi comprese le motivazioni che hanno portato alla scelta;
- **Pianificazione:** si rende necessario pianificare con attenzione le attività da eseguire nelle diverse fasi del progetto e le loro scadenze temporali. I titoli che vengono dati alle fasi definite nella pianificazione indicano le attività principali che il gruppo ha intenzione di svolgere in quella fase ma non le uniche che la impegneranno.
- **Preventivi e Consuntivo:** al fine di calcolare un preventivo vengono stimati i costi del lavoro di ogni attività. In seguito, verranno anche forniti i preventivi a finire ed i consuntivi al fine di calcolare le differenze tra il preventivo iniziale, il preventivo a finire ed il consuntivo per poter ricalcolare i successivi preventivi con una precisione maggiore.

2.1.1.3 Piano di Qualifica

Il *PianoDiQualifica_v.1.0.0*, redatto dai *Verificatori* ha come obiettivo la scelta delle strategie per la *qualifica_G* delle attività al fine di garantire la *qualità_G* del materiale prodotto dal gruppo. I contenuti principali del *PianoDiQualifica_v.1.0.0* saranno:

- **Qualità di processo:** si identificano le metriche per il controllo della qualità del processo;
- **Qualità di prodotto:** si identificano le metriche per il controllo della qualità del prodotto;
- **Specifiche dei Test:** qui vengono specificati dei test necessari a garantire che il prodotto soddisfi tutti i requisiti;
- **Standard di Qualità:** si delineano gli *standard di qualità_G* scelti;
- **Resoconto delle attività di verifica:** qui vengono riportati i risultati delle attività di *verifica_G* e di *validazione_G*;
- **Valutazioni per il Miglioramento:** al fine di un miglioramento continuativo nel tempo qui vengono riportate le problematiche relative al ricoprire un determinato ruolo o all'utilizzo di un determinato strumento, insieme con le relative soluzioni proposte.

2.1.2 Strumenti

Gli strumenti relativi alle attività del processo di fornitura sono descritti in seguito.

2.1.2.1 Microsoft Project

2.1.2.1.1 Descrizione Generale

Microsoft Project è un software di pianificazione sviluppato e venduto da Microsoft. È uno strumento per assistere i project manager nella pianificazione, nell'assegnazione delle risorse, nella verifica del rispetto dei tempi, nella gestione dei budget e nell'analisi dei carichi di lavoro.

2.1.2.1.2 Applicazione al Progetto

Ai fini del nostro progetto viene usato in particolar modo per la produzione di *Diagrammi di Gantt_G*.

2.1.2.2 Microsoft Excel**2.1.2.2.1 Descrizione Generale**

Microsoft Excel è un programma prodotto da Microsoft, dedicato alla produzione ed alla gestione di fogli elettronici. È parte della suite di software di produttività personale Microsoft Office, ed è disponibile per i sistemi operativi Windows e Macintosh.

2.1.2.2.2 Applicazione al Progetto

Il gruppo ha scelto di utilizzare Microsoft Excel per la produzione delle tabelle presenti nel *PianoDiProgetto_v.1.0.0*, dei grafici a torta e a barre.

2.2 Sviluppo

Questo processo contiene le attività ed i compiti necessari alla realizzazione del prodotto finale, in particolar modo questo processo deve tenere conto di:

- fissare gli obiettivi di sviluppo;
- fissare i vincoli tecnologici e di design;
- realizzare un prodotto finale che supera i test;
- realizzare un prodotto finale che soddisfa i requisiti e le richieste del proponente.

Seguendo le linee guida *ISO*_G \ *IEC*_G 12207:1995 abbiamo suddiviso il processo nelle seguenti attività:

- Analisi dei Requisiti;
- Progettazione;
- Codifica.

2.2.1 Attività

2.2.1.1 Analisi Dei Requisiti

Questo documento, redatto dagli analisti, ha lo scopo di riportare i requisiti del progetto in maniera formale.

Questi ultimi vengono estrapolati da diverse fonti:

- documenti di specifica del capitolato;
- fonti normative;
- casi d'uso;
- incontri con il proponente.

2.2.1.1.1 Casi d'uso

I casi d'uso rappresentano dei comportamenti offerti o desiderati espressi sotto forma di *diagrammi UML*_G. La loro struttura è composta da:

- codice identificativo univoco, nella forma **UC-x.y**, dove le componenti significano:
 - UC specifica che si tratta di un caso d'uso;
 - x è un numero progressivo che identifica univocamente il caso d'uso;
 - y è un numero progressivo che individua un sottocaso d'uso.
- titolo;
- attori;
- scopo;
- descrizione;
- precondizioni;
- postcondizioni;

- flusso degli eventi principali;
- scenario alternativo (se presente);
- estensioni (se presenti).

2.2.1.1.2 Requisiti

I requisiti rappresentano le specifiche che il sistema deve adempiere. Questi vengono estrapolati da diverse fonti e sono composti da:

- codice identificativo univoco nella forma R-X-T-I, dove:
 - **R** sta per requisito;
 - **X** è un carattere numerico ordinato crescente che identifica il requisito univocamente all'interno della sua tipologia di requisiti;
 - **T** è un carattere alfabetico che distingue la tipologia di requisito ovvero **F**unzionale, **Q**ualitativo, **D**ichiarativo;
 - **I** è un carattere alfabetico che segnala l'importanza di un requisito che può dunque essere **O**bligatorio, **D**esiderabile, **F**acoltativo.
- descrizione esaustiva;
- fonte dalla quale viene estratto.

2.2.1.2 Progettazione

Questo processo viene inizializzato dopo la prima stesura dell' *AnalisiDeiRequisiti_v.1.0.0*, la quale può essere o meno considerata definitiva a seconda del modello di ciclo di vita scelto per lo specifico progetto. A questo punto bisogna delineare le caratteristiche del prodotto, le quali devono venire approvate da tutti gli *stakeholders*_G. Quest'ultima attività verrà suddivisa essenzialmente in due parti:

- **Technology Baseline**: contiene le specifiche, di alto livello, che verranno poi usate nella realizzazione del sistema. Contiene inoltre i diagrammi UML che verranno poi utilizzati per la realizzazione dell'architettura e dei test di verifica;
- **Product Baseline**: stabilisce i test necessari alla verifica ed integra la Technology Baseline, dettagliandone ulteriormente l'attività.

2.2.1.3 Codifica

Al fine di una corretta gestione della qualità, i quali criteri verranno poi specificati nel *PianoDiQualifica_v.1.0.0*, si rende necessario normare la *codifica*_G. Ciò si rende necessario anche per:

- permettere la generazione di codice leggibile ed uniforme;
- migliorare la qualità del prodotto;
- agevolare le fasi di verifica, validazione e manutenzione del prodotto.

2.2.1.3.1 Stili di codifica

In questa sezione verrà delineato lo stile di codifica che il gruppo ha deciso di adottare:

- **indentazione:** i blocchi di testo devono venire correttamente indentati, con una indentazione di 4 caratteri spazio;
- **parentesizzazione:** le parentesi andranno messe "in linea" piuttosto che a capo;
- **nomi:** i nomi dovranno essere semplici ed esplicativi. Inoltre a seconda di ciò a cui si riferiscono i nomi valgono le seguenti regole:
 - per i nomi delle **classi** verrà seguita la notazione *CamelCase_G* (esempio: NomeClasse);
 - per i nomi delle **funzioni** verrà seguita la notazione *CamelCase*, con l'eccezione della prima lettera che andrà minuscola (esempio: nomeFunzione);
 - i nomi delle **costanti** verrà seguita la notazione *SnakeCase_G* e con tutte le lettere maiuscole (esempio: NOME_COSTANTE).

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo di questa sezione è definire gli standard riguardanti i documenti relativi ad ogni processo e attività significativi, volti allo sviluppo del progetto. I documenti sono consultabili nelle apposite sezioni del repository: <https://github.com/varTmp7/Stalker-Documents>.

3.1.2 Ciclo di vita del documento

Ogni documento segue le fasi del seguente ciclo di vita:

- **creazione del documento:** viene creato il documento in conformità alle norme adeguandolo a documenti precedenti dello stesso tipo, utilizzando il template reperibile nel repository;
- **realizzazione:** il documento viene scritto progressivamente, utilizzando un approccio incrementale;
- **in revisione:** il documento è prodotto interamente e rimane in attesa di verifica. Ogni documento viene verificato da almeno una persona, diversa dal redattore dello stesso.
- **approvato:** il responsabile di progetto dichiara che il documento è completo in ogni sua parte e pronto per essere rilasciato.

3.1.3 Template

Per facilitare la stesura dei documenti, è stato creato un template *L^AT_EX* che contiene la struttura grafica e lo stile di formattazione adottati. In questo modo, oltre a velocizzare la stesura permettendo ai redattori di concentrarsi solo sulla scrittura dei contenuti, il template permette di adottare automaticamente le conformità previste dalle *NormeDiProgetto_v.1.0.0*.

3.1.4 Struttura dei documenti

Ogni documento, ad eccezione della lettera di presentazione, deve essere realizzato secondo la seguente struttura.

3.1.4.1 Prima pagina

Il frontespizio è la prima pagina del documento ed è così formato:

- logo del gruppo: logo di *VarTmp7* posizionato in alto a centro pagina;
- gruppo e progetto: nome del gruppo e del progetto *Stalker*, visibili centralmente subito sotto il logo;
- titolo: nome del documento, posizionato centralmente in grassetto;
- tabella: presente sotto il titolo del documento, centrale e contenente le seguenti informazioni:
 - versione: versione del documento;
 - approvatori: nome e cognome dei membri del gruppo incaricati dell'approvazione del documento;

- redattori: nome e cognome dei membri del gruppo incaricati della redazione del documento;
- verificatori: nome e cognome dei membri del gruppo incaricati della verifica del documento;
- uso: tipo d'uso che può essere interno o esterno;
- distribuzione: destinatari del documento.
- descrizione: descrizione sintetica relativa al documento, centrale, posta sotto la tabella.

3.1.4.2 Registro delle modifiche

Ogni documento dispone di un *Registro delle Modifiche*, ovvero un *changelog_G* in formato tabellare che riporta le seguenti affermazioni:

- versione del documento dopo la modifica;
- data della modifica;
- descrizione sintetica della modifica apportata;
- nominativo di chi ha apportato la modifica;
- ruolo di chi ha apportato la modifica.

3.1.4.3 Indice

Lo scopo dell'indice è dare una visione d'insieme della struttura del documento, mostrando le parti gerarchiche che lo compongono. Ogni documento è composto dall'indice dei contenuti, posizionato dopo il *Registro delle Modifiche*.

3.1.4.4 Elenco immagini

I documenti muniti di figure, saranno provvisti di un indice delle figure contenente, per ciascuna, una breve descrizione e il numero della pagina in cui è collocata.

3.1.4.5 Elenco tabelle

I documenti muniti di tabelle, saranno provvisti di un indice delle tabelle contenente, per ciascuna, una breve descrizione e il numero della pagina in cui è collocata.

3.1.4.6 Contenuto principale

La struttura delle pagine di contenuto è la seguente:

- il logo del gruppo è posizionato in alto a sinistra;
- il numero e nome della sezione sono posizionati in alto a destra;
- una riga divide l'intestazione dal contenuto;
- tra l'intestazione e il piè di pagina troviamo il contenuto;
- una riga divide il contenuto dal piè di pagina;
- in basso a destra è presente il numero della pagina corrente.

3.1.5 Classificazione dei documenti

Tutti i documenti devono ricevere l'approvazione del *Responsabile* prima della loro divulgazione.

3.1.5.1 Documenti ad uso interno

Sono da considerarsi documenti ad uso interno al gruppo:

- tutti i *Verbali interni*;
- lo *StudioDiFattibilità_v.1.0.0*;
- le *NormeDiProgetto_v.1.0.0*.

3.1.5.2 Documenti ad uso esterno

I documenti destinati ad essere divulgati all'esterno del gruppo sono:

- l'*AnalisiDeiRequisiti_v.1.0.0*;
- il *PianoDiQualifica_v.1.0.0*;
- il *PianoDiProgetto_v.1.0.0*;
- la *Lettera di presentazione*;
- il *Glossario_v.1.0.0*.

3.1.5.3 Verbali

I verbali vengono prodotti da un incaricato che avrà il ruolo di *Redattore*, ad ogni incontro interno al gruppo o con i proponenti del progetto. Per i verbali è prevista un'unica stesura, in quanto una modifica successiva all'incontro implicherebbe un cambio nelle decisioni prese in modo retroattivo. Oltre alla struttura standard di tutti gli altri documenti descritta in precedenza, il verbale è costituito nelle pagine di contenuto da una sezione di *Informazioni generali*, che contengono nell'ordine:

- luogo di svolgimento della riunione;
- data nella quale si è svolta;
- ora di inizio;
- ora di fine;
- membri del gruppo/esterni presenti.

Segue l'*Ordine del giorno*, una breve descrizione degli argomenti da discutere nella riunione. Successivamente è presente il *Verbale della riunione*, una descrizione più dettagliata, sotto forma di elenco puntato, di tutti gli argomenti discussi e delle decisioni prese durante l'incontro. Ogni verbale dovrà essere denominato secondo il seguente formato

Verbale_X_YYYY-MM-DD

dove per *X* si intende il tipo di verbale, che sarà:

- **I**: se l'incontro è stato interno al gruppo;
- **E**: se l'incontro si è svolto con la partecipazione dei proponenti.

In conformità allo standard ISO 8601, le date devono essere scritte secondo il formato: *YYYY-MM-DD*, dove:

- **YYYY**: rappresenta l'anno con 4 cifre;
- **MM**: rappresenta il mese con 2 cifre;
- **DD**: rappresenta il giorno con 2 cifre.

Questo formato verrà utilizzato in tutti i documenti, ad eccezione dei diagrammi creati con *MS Project*, le cui date saranno scritte nel modo seguente: *YY-MM-DD*. Gli orari invece, sempre secondo lo stesso standard, devono essere scritti nel formato *HH:MM*, dove:

- **HH**: rappresenta l'ora con 2 cifre in formato 24 ore;
- **MM**: rappresenta i minuti con 2 cifre.

3.1.6 Norme tipografiche

3.1.6.1 Nomi dei file

I nomi dei file (estensione esclusa) e delle cartelle utilizzano la convenzione *CamelCase* e alcune regole aggiuntive elencate di seguito:

- i nomi dei file e delle cartelle avranno sempre come lettera iniziale una lettera maiuscola;
- i nomi composti verranno scritti unendo le parole tra loro e lasciando le loro iniziali maiuscole;
- le preposizioni non si omettono;
- i nomi dei documenti includeranno la versione e seguiranno il formato: *NomeDelFile_v.X.Y.Z*, dove X.Y.Z indica la versione attuale del documento.

Un esempio corretto è: *NormeDiProgetto_v.1.0.0*.

3.1.6.2 Separatore migliaia

Per facilitare la lettura dei numeri con più di tre cifre intere, si è deciso di fare riferimento allo standard internazionale ISO 31-0, che raccomanda l'uso dello spazio per raggruppare le cifre. Un esempio corretto è: *12 320*, che indica il numero *dodicimilatrecentoventi*.

3.1.6.3 Elenchi puntati

Gli elenchi puntati presenti nei documenti rispetteranno il seguente formato:

- la frase che precede il primo elemento dell'elenco deve terminare con il carattere ":";
- se un elemento dell'elenco a sua volta precede un sottoelenco, dovrà terminare con il carattere ":";
- ogni elemento dell'elenco deve cominciare con la lettera minuscola, tranne quando è un nome proprio o una parola da glossario;
- ogni elemento dell'elenco, se è costituito da un termine o da un solo periodo, deve terminare con il carattere ";" altrimenti se è costituito da più periodi, divisi tra loro dal carattere ".", l'elemento terminerà con il carattere ".";
- l'ultimo elemento dell'elenco terminerà in ogni caso con il carattere ".".

3.1.6.4 Glossario

I termini del *Glossario_v.1.0.0* presenti nel testo dei documenti sono marcati con una G maiuscola a pedice alla loro prima occorrenza e scritti in corsivo. Se nel *Glossario_v.1.0.0* un termine presenta una descrizione che utilizza termini da glossario, è necessario utilizzare il comando `\LATEX` appositamente creato che formatta la parola in corsivo e con una G maiuscola a pedice. Le parole da glossario presenti nei titoli o nelle didascalie di immagini e tabelle non vengono segnalate. Solo la prima occorrenza della parola da glossario presente nel corpo del documento viene segnalata. Se il termine da inserire nel *Glossario_v.1.0.0* è composto da più parole, tutte le parole comprese le preposizioni saranno scritte in corsivo. Un esempio corretto è: *ciclo di vita_G*.

3.1.6.5 Sigle

In tutti i documenti verranno utilizzate le seguenti sigle per abbreviare i nomi di documenti da citare nel testo, con l'esclusione dei titoli di sezione o paragrafo:

- **AdR**: indica l'*Analisi Dei Requisiti_v.1.0.0*;
- **NdP**: indica le *Norme Di Progetto_v.1.0.0*;
- **PdP**: indica il *Piano Di Progetto_v.1.0.0*;
- **PdQ**: indica il *Piano Di Qualifica_v.1.0.0*;
- **SdF**: indica lo *Studio Di Fattibilità_v.1.0.0*.

Le seguenti sigle invece sono utilizzate per abbreviare i nomi delle varie revisioni, verranno utilizzate nei nomi delle cartelle e nei documenti:

- **RR**: indica la *Revisione dei Requisiti*;
- **RP**: indica la *Revisione della Progettazione*;
- **RQ**: indica la *Revisione di Qualifica*;
- **RA**: indica la *Revisione di Accettazione*.

3.1.7 Componenti grafiche

3.1.7.1 Tabelle

Di seguito sono riportate le norme da utilizzare per lo stile delle tabelle che compaiono nei diversi documenti.

- l'*header_G* della tabella ha lo sfondo blu e il colore del testo bianco;
- la prima parola di ogni cella ha l'iniziale maiuscola;
- lo sfondo delle righe di contenuto della tabella è di colore bianco alternato a grigio;
- la tabella è centrata al documento.

Di seguito troviamo un esempio corretto.

Parametro 1	Parametro 2
Contenuto 1	Questa è la prima riga di contenuto
Contenuto 2	Questa è la seconda riga di contenuto

Tabella 2: Esempio corretto di tabella.

3.1.7.2 Immagini

Le immagini devono essere messe in una cartella, dentro alla cartella del documento a cui appartengono, denominata *img*. Il formato usato per nominare i file immagine è il seguente: *S_ss_#_nome.estensione*, dove:

- **S** è il numero di sezione in cui verrà inserita l'immagine;
- **ss** è il numero di sottosezione in cui verrà inserita l'immagine;
- **#** è il numero progressivo dell'immagine all'interno della sottosezione;
- **nome** è un nome significativo per l'immagine;
- **estensione** è il formato dell'immagine.

Tutte le immagini saranno centrate e seguite da una breve descrizione.

3.1.7.3 Diagrammi UML

I diagrammi UML presenti nei vari documenti, andranno inseriti in formato immagine *png*.

3.1.8 Strumenti

3.1.8.1 L^AT_EX

Lo strumento utilizzato per la scrittura dei documenti è *L^AT_EX*, un linguaggio di markup per la preparazione di testi, basato sul programma di composizione tipografica *Tex*.

3.1.8.2 Astah UML

Per la produzione dei diagrammi UML è stato scelto di utilizzare *Astah UML*, un editor gratuito e intuitivo.

3.2 Gestione della configurazione

3.2.1 Versionamento

3.2.1.1 Codifica della versione del documento

Ogni documento può essere ricostruito attraverso le sue versioni, di cui si tiene traccia nel registro delle modifiche, dove ogni riga corrisponde ad una versione. Ogni versione del documento viene codificata con il formato *X.Y.Z*, dove:

- **X** rappresenta una versione completa del documento, verificata e rilasciata dal responsabile:
 - parte da 0;
 - viene incrementato dal responsabile di progetto al rilascio.
- **Y** rappresenta una versione la cui stesura è stata completata ma rimane in attesa di essere verificata:
 - parte da 0;
 - viene incrementato dal verificatore a fine verifica;
 - all'incremento di X riparte da 0.
- **Z** rappresenta una versione ancora in fase di stesura:
 - parte da 0;
 - viene incrementato dal redattore ad ogni modifica consistente;
 - all'incremento di Y riparte da 0.

3.2.2 Repository

3.2.2.1 Tecnologie e strumenti

Per il progetto è stato utilizzato il sistema di controllo versione distribuito di *Git*, usando il servizio di *Github* per ospitare il repository. Per facilitarne l'utilizzo, si è scelto di fare uso del software *GitKraken*, una valida alternativa alla riga di comando che offre anche un buon sistema di issue tracking chiamato *Glo Boards*.

3.2.2.2 Struttura del repository

Ogni membro del gruppo ha in locale una copia del repository, clonata dal principale ospitato in remoto. L'organizzazione delle cartelle è la seguente:

- **RR**: cartella che contiene i documenti richiesti alla revisione dei requisiti, strutturata in questo modo:
 - **Documenti interni**: cartella che contiene i *Verbali interni*, le *NdP* e lo *SdF*;
 - **Documenti esterni**: cartella che contiene i *Verbali esterni*, l'*AdR*, il *PdP*, il *PdQ* e il *Glossario*;
 - **Lettera di presentazione**.
- **TemplateDocumento**: cartella che contiene i file necessari a creare un nuovo documento e un file di comandi personalizzati per facilitarne la stesura.

3.2.2.3 Utilizzo di GitKraken

Il gruppo ha deciso di suddividere nella prima fase il repository in vari branch, uno per ogni documento da creare. Ogni membro del gruppo seguirà questa procedura:

- scelto il branch in cui lavorare sulla base del documento a lui assegnato, eseguire un *pull_G*, in modo da aggiornare la sua copia locale con le ultime modifiche;
- svolto il lavoro assegnato, confermare le modifiche apportate nell'area di *staging_G*;
- eseguire il *commit_G*, lasciando un breve commento che ne descriva il contenuto;
- effettuare il *push_G* delle modifiche, in modo da aggiornare il repository in remoto.

3.2.2.4 Gestione delle modifiche

Tutti i membri del team possono modificare tutti i file in ogni branch. Per modifiche di tipo grammaticale od ortografico, si contattterà direttamente il *redattore* tramite il canale *Slack* relativo al documento interessato. Quest'ultimo si occuperà di apportare la correzione. Per modifiche di maggiore entità, si contatterà sempre il *redattore* tramite *Slack* e si aprirà un'issue assegnandola a se stessi e al redattore, con una descrizione il più possibile precisa in modo da facilitare il lavoro di correzione.

3.3 Gestione della qualità

Lo scopo è fornire un'adeguata garanzia che i prodotti e i processi software rispettino gli obiettivi di qualità, aderiscano ai piani stabiliti nel ciclo di vita del software e che i bisogni del proponente siano soddisfatti. Per la trattazione approfondita della gestione della qualità si fa riferimento al *PdQ*, nel quale vengono descritte le modalità di perseguimento della qualità. Nello specifico troviamo:

- gli standard utilizzati;
- i processi di interesse negli standard;
- gli attributi del software più significativi per il progetto.

Per ogni processo vengono descritti:

- gli obiettivi;
- la strategia;
- le metriche.

Le caratteristiche che rendono un prodotto di qualità sono descritte attraverso:

- gli obiettivi;
- le metriche.

3.3.1 Strumenti

Gli strumenti predefiniti per la qualità sono:

- gli standard (*ISO/IEC/IEEE-12207:1995* per i processi e *ISO/IEC 9126* per i prodotti);
- le metriche descritte nel *PdQ*.

3.4 Verifica

Il processo di verifica ha lo scopo di individuare gli errori introdotti nel prodotto durante la fase di redazione o sviluppo in modo da permettere la realizzazione di prodotti corretti, coesi e completi. Sono soggetti a verifica sia il software che i documenti. Per ottenere questo risultato, il processo si affida alle attività di analisi e di test.

3.4.1 Attività

3.4.1.1 Analisi

Il processo di analisi si divide in due metodi: analisi statica e analisi dinamica.

3.4.1.1.1 Analisi Statica

L'analisi statica studia la documentazione e il codice, accertandone la conformità alle regole, l'assenza di difetti e la presenza delle proprietà desiderate. Non richiede l'esecuzione del codice, quindi è attuabile già nelle prime fasi. Esistono due tecniche per effettuare analisi statica, che sono:

- **Walkthrough:** tecnica che consiste in una lettura a largo spettro in cerca di anomalie. Verrà utilizzata nella fase iniziale del progetto non avendo ancora una profonda conoscenza delle NdP e del PdQ;
- **Inspection:** tecnica che consiste in una lettura mirata e strutturata, utilizzando le *liste di controllo*_G e le misurazioni effettuate in precedenza, in modo da rendere sempre più efficace la tecnica con l'esperienza.

Segue una prima lista di controllo per la verifica dei documenti. Se ne prevede l'ampliamento.

Oggetto	Descrizione
Formato data	La data deve rispettare il formato <i>YYYY-MM-DD</i>
Sintassi	Frase semplificabile o errore nella sua struttura
Elenchi puntati	Uso della punteggiatura non conforme alle norme
Glossario	Termine già presente segnalato da inserire nel Glossario

Tabella 3: Lista di controllo per la verifica

3.4.1.1.2 Analisi Dinamica

L'analisi dinamica viene effettuata tramite dei test che verificano il corretto funzionamento del prodotto software e rilevano eventuali errori. Richiede l'esecuzione del prodotto software, quindi è attuabile solamente dopo l'attività di codifica.

3.4.1.2 Test

3.4.1.2.1 Test di unità

I test di unità si eseguono su singole unità di software e permettono di verificarne la correttezza.

za. Le singole unità possono essere testate con l'ausilio di *driver_G* e *stub_G* che rispettivamente simulano un'unità chiamante e un'unità chiamata.

3.4.1.2.2 Test di integrazione

I test di integrazione sono l'estensione dei test di unità. Si concentrano sulle interfacce tra i componenti. Al superamento dei test si formano agglomerati sempre più grandi di unità, fino ad arrivare alla dimensione totale del sistema.

3.4.1.2.3 Test di sistema

Il test di sistema viene fatto alla validazione del prodotto finale, una volta integrati tutti i suoi componenti. Ne verifica il comportamento dinamico rispetto ai requisiti software.

3.4.1.2.4 Test di regressione

Il test di regressione viene effettuato a seguito di una modifica per correzione o estensione di parti, in modo da accertare che non comportino errori nel sistema.

3.4.1.2.5 Test di accettazione

Il test di accettazione si effettua con la presenza del committente e serve ad accertare il soddisfacimento dei requisiti utente. Il suo superamento garantisce che il software è pronto per essere rilasciato.

3.4.2 Strumenti

3.4.2.1 Verifica ortografica

La verifica ortografica viene effettuata nei plugin offerti dagli editor di studio utilizzati dal gruppo, *TexStudio*, *JEdit* e *Overleaf*.

3.4.2.2 Verifica codice

Si fa affidamento allo strumento offerto da W3C per la validazione del codice di markup *HTML5* e *CSS3*, disponibile al link <https://validator.w3.org/>. Verranno adottati altri strumenti successivamente.

3.4.2.3 Indice di Gulpease

È stato creato uno script in *Python* che, dato in input un file in formato *pdf*, restituisce l'*indice di Gulpease_G*.

3.5 Validazione

Lo scopo della validazione è accertare l'efficacia del prodotto finale. L'esito positivo del processo, garantisce che il software rispetti tutti i requisiti e soddisfi i bisogni del committente.

3.5.1 Attività

Le attività da svolgere per validare il prodotto sono:

- identificazione degli oggetti da validare;
- decisione di una strategia che permetta un riutilizzo delle procedure di verifica;
- applicazione della strategia, tenendo traccia dei risultati ottenuti;
- valutazione dei risultati ottenuti in confronto alle aspettative iniziali.

4 Processi organizzativi

4.1 Gestione organizzativa

4.1.1 Ruoli di progetto

Ogni membro del gruppo ricoprirà, a rotazione, i ruoli di progetto di seguito descritti. Nel *PdP* verranno assegnate e organizzate le attività relative ai vari ruoli.

4.1.1.1 Responsabile di progetto

Il responsabile di progetto ha il compito di gestire le risorse umane, i rischi, la pianificazione, il controllo, il coordinamento e le relazioni esterne. Inoltre, ha il compito di fare da intermediario tra il gruppo e le persone esterne; competono a lui quindi le comunicazioni con committente e proponente.

4.1.1.2 Amministratore di progetto

L'amministratore ha il compito di supportare e controllare l'ambiente di lavoro. Si occupa quindi di gestire il controllo della configurazione del prodotto, del versionamento e della documentazione.

4.1.1.3 Analista

L'analista si occupa dell'attività di analisi del problema, definendone i requisiti espliciti, impliciti e del dominio applicativo. Ha il compito di redigere lo *SdF* e l'*AdR*. Questa figura non sarà sempre presente durante il progetto.

4.1.1.4 Progettista

Il progettista si occupa degli aspetti tecnologici e tecnici del progetto. Ha il compito di effettuare lo *SdF* del prodotto, sviluppare un'architettura efficiente ed efficace sulla base del lavoro dell'analista e di redigere la specifica tecnica del progetto.

4.1.1.5 Verificatore

Il verificatore ha il compito di gestire le attività di verifica e validazione, durante l'intero ciclo di vita del progetto. Si occupa di verificare la correttezza della documentazione e del codice affidandosi alle norme definite nelle *NdP*, segnalando eventuali errori alle persone che hanno in carico l'oggetto in discussione.

4.1.1.6 Programmatore

Il programmatore ha il compito di produrre il codice necessario allo sviluppo del prodotto e alle componenti di supporto utili alla verifica e validazione. Deve attenersi alle specifiche fornite dal responsabile e ha il compito di produrre codice che sia facilmente mantenibile.

4.1.2 Procedure

In questa sezione vengono descritte le norme che regolano le comunicazioni tra i membri del gruppo e con le parti esterne coinvolte.

4.1.2.1 Gestione delle comunicazioni

Il gruppo utilizza un *workspace_G* di *Slack* per comunicare internamente. Sono presenti diversi canali, uno per ogni documento, in aggiunta di un canale generale per informazioni utili a tutto il gruppo e ad un canale riservato alle comunicazioni con un referente dei proponenti, *Imola*

Informatica. Per comunicazioni e informazioni di carattere organizzativo interne al gruppo, viene utilizzato un canale *Telegram*.

4.1.2.2 Gestione degli incontri

4.1.2.2.1 Incontri interni

Gli incontri interni al gruppo sono organizzati dal responsabile, che si occupa di scegliere in accordo con gli altri membri, data e orario dell'incontro sulla base della disponibilità di quest'ultimi, resa pubblica in un documento condiviso in *Google Drive*, aggiornato settimanalmente.

4.1.2.2.2 Incontri esterni

Gli incontri esterni sono, allo stesso modo, organizzati dal responsabile. Quando si presenta la necessità di un incontro di questo tipo, sia per volontà di uno o più membri del gruppo, del proponente o del committente, il responsabile si occupa di comunicare una data e un orario deciso in accordo con le parti, comunicandolo attraverso i canali precedentemente citati.

4.1.3 Formazione

4.1.3.1 Conoscenze pregresse

Di seguito sono elencate le varie conoscenze, affini al progetto, dichiarate dai membri del gruppo:

- **Stefano Cavaliere:** C++, Java, HTML, CSS, Bootstrap, SQL;
- **Marco Ferrati:** C++, Swift, iOS, Python, Java, Git, SQL;
- **Lorenzo Taschin:** C++, Java, Python, Javascript, HTML, SQL;
- **Riccardo Tassetto:** C++, C#, Java, CSS, HTML, PHP, Bootstrap, SQL, Git;
- **Marcel Junior Wandji:** C++, CSS, Git, HTML, Java, Javascript, SQL, PHP;
- **Xiaowei Wen:** Android (in Java), Java, HTML, Python, C++, Git, MS Excel, SQL, MS Project;
- **Giuseppe Zatta:** C++, Java, HTML, CSS, PHP, MS Office, Git, SQL;
- **Claudia Zattara:** C++, Java, HTML, CSS, Javascript, PHP, SQL, Git, MS Office.

È prevista una formazione autonoma da parte di tutti i membri del gruppo, in modo da sfruttare al meglio le tecnologie che verranno utilizzate durante il progetto. Per facilitare la formazione i membri del gruppo sono tenuti a condividere, oltre alle loro conoscenze, eventuali materiali utili caricandoli nell'apposita cartella condivisa di Google Drive.

4.1.4 Strumenti di coordinamento

4.1.4.1 Glo Boards

Come accennato in precedenza, *Glo Boards* è uno strumento fornito da *GitKraken* per la gestione delle issues. Permette a tutti i membri del gruppo di creare nuove issues, assegnandole agli interessati e notificando le relative scadenze. Lo strumento consiste in una lavagna virtuale dove sono visibili i vari tasks creati, associati ad una data di scadenza e assegnati ad uno o più membri. Gli stati in cui si possono trovare i tasks sono i seguenti:

- **To do:** il lavoro deve ancora essere preso in carico;
- **In progress:** i membri assegnati stanno lavorando a questo compito;

- **Verify:** il lavoro svolto è terminato ed è sotto verifica;
- **Done:** il task è stato completato.

La scelta è ricaduta su questo strumento per l'intuitività dell'interfaccia e la facilità di utilizzo.