



8Lab Solutions - Progetto "Soldino"

Norme di Progetto

Versione	1.0.0
Approvazione	
Redazione	
Verifica	
Stato	
Uso	Interno
Destinato a	8Lab Solutions Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Questo documento racchiude le regole, gli strumenti e le convenzioni adottate dal gruppo 8Lab Solutions durante la realizzazione del progetto Soldino.

8labsolutions@gmail.com

Changelog

Versione	Data	Nominativo	Ruolo	Descrizione
0.0.3	26/11/2018	Federico Biciato	<i>Analista</i>	Stesura § 2.
0.0.2	26/11/2018	Paolo Pozzan	<i>Revisore</i>	Stesura § 1.
0.0.1	25/11/2018	Mattia Bolzonella	<i>Analista</i>	Creato documento latex e creata pagina del titolo.

Tabella 1: Changelog di questo documento

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Ambiguità	5
1.4	Riferimenti	5
1.4.1	Riferimenti normativi	5
1.4.2	Riferimenti informativi	5
2	Processi primari	7
2.1	Acquisizione	7
2.1.1	Scopo del processo	7
2.1.2	Descrizione	7
2.1.3	Strumenti	7
2.1.3.1	Skype, Hangouts	7
2.1.3.2	Slack	7
2.2	Fornitura	7
2.2.1	Scopo del processo	7
2.2.2	Aspettative	8
2.2.3	Descrizione	8
2.2.4	Attività	8
2.2.4.1	Studio di fattibilità	8
2.2.4.2	Piano di progetto	9
2.2.4.3	Piano di qualifica	9
2.3	Sviluppo	9
2.3.1	Scopo del processo	9
2.3.2	Aspettative	9
2.3.3	Descrizione	10
2.3.4	Attività	10
2.3.4.1	Analisi dei Requisiti	10
2.3.4.1.1	Scopo	10
2.3.4.1.2	Aspettative	10
2.3.4.1.3	Descrizione	10
2.3.4.1.4	Casi d'uso	10
2.3.4.1.5	Codice identificativo dei casi d'uso	11
2.3.4.1.6	Requisiti	11
2.3.4.1.7	Codice identificativo dei requisiti	11
2.3.4.1.8	UML	12
2.3.4.2	Progettazione	12
2.3.4.2.1	Scopo	12
2.3.4.2.2	Aspettative	12
2.3.4.2.3	Descrizione	12
2.3.4.2.4	Specifica tecnica	12
2.3.4.2.5	Definizione di prodotto	13
2.3.4.3	Codifica	13
2.3.4.3.1	Scopo	13
2.3.4.3.2	Aspettative	13

2.3.4.3.3	Descrizione	13
2.3.4.3.4	Stile di codifica	13
2.3.4.3.5	Ricorsione	14
2.3.5	Strumenti	14
2.3.5.1	Draw.io	14
2.3.5.2	IntelliJ IDEA	14
3	Processi di Supporto	15
3.1	Documentazione	15
3.1.1	Scopo	15
3.1.2	Aspettative	15
3.1.3	Descrizione	15
3.1.4	Ciclo di vita del documento	15
3.1.5	Procedure	15
3.1.5.1	Approvazione dei documenti	16
3.1.6	Template	16
3.1.7	Struttura dei documenti	16
3.1.7.1	Prima pagina	16
3.1.7.2	Registro modifiche	17
3.1.7.3	Indice	17
3.1.7.4	Contenuto principale	17
3.1.7.5	Note a piè pagina	17
3.1.8	Versionamento	18
3.1.8.1	Processo di versionamento	18
3.1.8.2	Codice per la versione	18
3.1.9	Norme tipografiche	18
3.1.9.1	Convenzioni sui nomi dei file	18
3.1.9.2	Stile del testo	19
3.1.9.3	Elenchi puntati	19
3.1.9.4	Formati comuni	19
3.1.9.5	Sigle	19
3.1.10	Elementi grafici	20
3.1.10.1	Changelog	20
3.1.10.2	Tabelle	20
3.1.10.3	Immagini	20
3.1.11	Classificazione dei documenti	20
3.1.11.1	Documenti informali	20
3.1.11.2	Documenti formali	21
3.1.11.3	Verbali	21
3.1.12	Strumenti	21
3.1.12.1	L ^A T _E X	21
3.1.12.2	TexStudio	22
3.1.12.3	Draw.io	22
3.2	Qualità	22
3.3	Verifica	22
3.3.1	Scopo	22
3.3.2	Aspettative	22
3.3.3	Descrizione	23
3.3.4	Attività	23

3.3.4.1	Analisi	23
3.3.4.1.1	Analisi statica	23
3.3.4.1.2	Analisi dinamica	23
3.3.4.2	Test	24
3.3.4.2.1	Test di unità	24
3.3.4.2.2	Test di integrazione	24
3.3.4.2.3	Test di sistema	24
3.3.4.2.4	Test di regressione	24
3.3.4.2.5	Test di collaudo	24
3.3.5	Strumenti	24
3.3.5.1	Verifica ortografica	24
3.3.5.2	Validazione W3C	25
4	Processi Organizzativi	26
4.1	Gestione Organizzativa	26
4.1.1	Scopo	26
4.1.2	Aspettative	26
4.1.3	Descrizione	26
4.1.4	Ruoli di progetto	26
4.1.4.1	Responsabile di progetto	26
4.1.4.2	Amministratore di progetto	27
4.1.4.3	Analista	27
4.1.4.4	Progettista	27
4.1.4.5	Programmatore	27
4.1.4.6	Verificatore	28
4.1.5	Procedure	28
4.1.5.1	Gestione delle comunicazioni	28
4.1.5.1.1	Comunicazioni interne	28
4.1.5.1.2	Comunicazioni esterne	28
4.1.5.2	Gestione degli incontri	28
4.1.5.2.1	Incontri interni del team	28
4.1.5.2.2	Verbali di riunioni interne	28
4.1.5.2.3	Incontri esterni del team	29
4.1.5.2.4	Verbali esterni di riunione	29
4.1.5.3	Gestione degli strumenti di coordinamento	29
4.1.5.3.1	Ticcking	29
4.1.5.4	Gestione degli strumenti di versionamento	29
4.1.5.4.1	Repository	29
4.1.5.4.2	Struttura del repository	29
4.1.5.4.3	Tipi di file e .gitignore	30
4.1.5.4.4	Lavoro collaborativo e versionamento	30
4.1.5.5	Gestione dei rischi	31
4.1.5.5.1	Codifica dei rischi	31
4.1.5.6	Gestione di progetto	31
4.1.6	Strumenti utilizzati	31
4.2	Formazione del team	32

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di definire delle regole di base che tutti i membri di 8LabSolutions devono rispettare nello svolgimento del progetto, così da garantire uniformità in tutto il materiale. Verrà utilizzato un approccio incrementale, volto a normare passo passo ogni decisione discussa e concordata tra tutti i membri del gruppo. Ogni eventuale modifica sarà in accordo fra tutti i componenti allo scopo di perseguire la coesione all'interno del team.

1.2 Scopo del prodotto

Il capitolato C6 si prefigge lo sviluppo di una piattaforma gestita dal governo, basata sull'infrastruttura Ethereum e funzionante tramite il meccanismo degli smart contracts. Soldino ha lo scopo di assistere le aziende nella gestione dell'IVA riguardante tutte le operazioni di compravendita di beni e servizi, e della gestione delle tasse. Il governo potrà coniare e distribuire la moneta utilizzata in queste transazioni, mentre i cittadini potranno acquistare i beni utilizzando tale valuta. La piattaforma, dunque, intende divenire un punto di incontro tra governo, aziende e cittadini.

1.3 Ambiguità

All'interno dei documenti alcuni termini presentano significati ambigui a seconda del contesto, fraintendibili, o che necessitano di una descrizione più approfondita. Per evitare questa ambiguità è stato creato il documento "Glossario" volto a fare chiarezza, ponendo a fianco di ogni termine il suo preciso significato. Questi termini sono pertanto marchiati con una "G" a pedice per ogni loro occorrenza all'interno di tutti i documenti.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Standard ISO/IEC 12207:1995**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;
- **Testo del capitolato**
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C6.pdf>;

1.4.2 Riferimenti informativi

- Piano di progetto;
- Piano di qualifica;
- Amministrazione di progetto - Slide del corso "Ingegneria del Software":
<https://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L05.pdf>;
- Guide to the Software Engineering Body of Knowledge(SWEBOK), 2004:
<http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf>;

- Software Engineering - Ian Sommerville - 9 th Edition (2010):
(formato cartaceo);
- Documentazione git:
<https://www.atlassian.com/git>;
- Standard ISO 8601:
https://it.wikipedia.org/wiki/ISO_8601

2 Processi primari

2.1 Acquisizione

2.1.1 Scopo del processo

Il processo di acquisizione definisce le attività dell'acquirente, dell'organizzazione che acquista il prodotto. Queste attività interessano tutto il periodo di realizzazione dello stesso. Il processo si articola in:

- avvio;
- preparazione di una presentazione del proprio capitolato;
- esposizione della presentazione agli studenti dell'università;
- preparazione di un contratto e aggiornamento dello stesso;
- primo incontro con i gruppi che si sono aggiudicati il capitolato d'appalto; conoscenza dei membri del gruppo e prima discussione su richieste e chiarimenti;
- monitoraggio periodica dell'avanzamento del lavoro sul progetto;
- accettazione del prodotto finale.

2.1.2 Descrizione

L'azienda si impegna a prendere visione periodicamente del lavoro degli studenti, organizzando incontri sia fisicamente, sia tramite chiamate online, per discutere determinati aspetti del progetto e per verificarne l'avanzamento.

2.1.3 Strumenti

Di seguito gli strumenti utilizzati dall'acquirente per mettersi in contatto col gruppo.

2.1.3.1 Skype, Hangouts

Software di messaggistica istantanea e di VoIP utilizzato per le chiamate online.

2.1.3.2 Slack

Strumenti di collaborazione aziendale utilizzato per inviare messaggi in modo istantaneo ai membri del team.

2.2 Fornitura

2.2.1 Scopo del processo

Il processo di fornitura si compone delle attività e dei compiti del fornitore. Una volta comprese le richieste del proponente e aver stilato uno studio di fattibilità, può essere avviato questo processo con fine di soddisfare ognuna di queste richieste. D'altra parte si deve stipulare e concordare con l'acquirente un contratto per la consegna del prodotto. Il processo precedentemente avviato continua con la determinazione delle procedure e delle risorse necessarie per il completamento del progetto, incluso lo sviluppo di un piano di progetto e l'esecuzione di questo fino alla consegna del materiale prodotto. Il processo di fornitura è composto dalle seguenti fasi:

- avvio;
- approntamento di risposte alle richieste;
- contrattazione;
- pianificazione;
- esecuzione e controllo;
- revisione e valutazione;
- consegna e completamento;

2.2.2 Aspettative

Il gruppo intende mantenere un costante dialogo con il proponente per avere un riscontro efficace sul lavoro svolto e instaurare un rapporto di collaborazione in termini di:

- determinare aspetti chiave per far fronte ai bisogni del proponente;
- stilare requisiti e vincoli sui processi;
- stimare le tempistiche di lavoro;
- promuovere una verifica continua;
- chiarire eventuali dubbi emersi;
- accordarsi sulla qualifica del prodotto.

2.2.3 Descrizione

Questa sezione tratta le norme che i membri del gruppo 8Lab Solutions devono rispettare in tutte le fasi di progettazione, sviluppo e consegna del prodotto Soldino, al fine di diventare fornitori nei confronti del proponente Red Babel e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin.

2.2.4 Attività

2.2.4.1 Studio di fattibilità

E' compito del responsabile di progetto organizzare riunioni tra i membri del gruppo al fine di permettere lo scambio di opinioni sui capitoli proposti. Il documento "Studio di Fattibilità", redatto dagli analisti, indica per ogni capitolo:

- **Descrizione e obiettivo finale:** viene fatta una presentazione del capitolo in generale, una descrizione delle caratteristiche principali richieste per il prodotto e viene definito l'obiettivo che si vuole raggiungere;
- **Studio del dominio:** vengono individuati i principali attori che saranno poi coinvolti, facenti parte del dominio applicativo e viene fatto un elenco delle tecnologie richieste per lo svolgimento, che rientrano nel dominio tecnologico;
- **Aspetti positivi e di rischio:** vengono espone le considerazioni fatte dal gruppo sugli aspetti positivi e sui fattori di rischio del capitolo;
- **Conclusioni:** vengono espone le ragioni per la quale il gruppo ha deciso di accettare o scartare il capitolo.

2.2.4.2 Piano di progetto

Il responsabile, con l'aiuto degli amministratori, redigerà un piano da seguire durante lo svolgimento del progetto, questo documento conterrà:

- **Analisi dei rischi:** vengono analizzati nel dettaglio i rischi che potranno presentarsi e viene fornita una possibile soluzione. Viene anche fornita la probabilità con la quale questi possono presentarsi e il livello di gravità per ciascuno;
- **Modello di sviluppo:** viene descritto il modello di sviluppo che è stato scelto;
- **Pianificazione:** vengono pianificate le attività da eseguire nelle diverse fasi del progetto e vengono stabilite le loro scadenze temporali;
- **Preventivo e consuntivo:** viene data una stima di lavoro necessaria per ciascuna fase proponendo così un preventivo per il costo totale del progetto. Verrà anche tracciato, alla fine di ogni attività, un consuntivo di periodo relativo all'andamento rispetto a ciò che è stato preventivato.

2.2.4.3 Piano di qualifica

I verificatori dovranno redigere un documento contenente le strategie da adottare per verifica e validazione del materiale che sarà prodotto dal gruppo. Quest'ultimo conterrà:

- **Strategia di gestione della qualità:** vengono definiti gli obiettivi di qualità di processo e di prodotto e individuate le metriche e le relative misure;
- **Gestione amministrativa della revisione:** vengono stabiliti gli obiettivi da raggiungere per la qualità di processo e di prodotto, assegnando un range alle metriche;
- **Standard di qualità:** vengono descritti gli standard di qualità scelti dal gruppo;
- **Valutazioni per il miglioramento:** vengono riportati i problemi e le relative soluzioni nel ricoprire un determinato ruolo e nell'uso degli strumenti scelti;
- **Resoconto delle attività di verifica:** per ogni attività si devono riportare le metriche calcolate e un resoconto sulla verifica di tale attività.

2.3 Sviluppo

2.3.1 Scopo del processo

Attività e compiti svolti dal gruppo, al fine di arrivare al prodotto finale richiesto dal proponente, sono contenuti in questo processo.

2.3.2 Aspettative

Per una corretta implementazione di tale processo le aspettative sono le seguenti:

- realizzare un prodotto finale conforme alle richieste del proponente;
- realizzare un prodotto finale che soddisfa i test di verifica;
- realizzare un prodotto finale che soddisfa i test di validazione;
- fissare gli obiettivi di sviluppo;
- fissare i vincoli tecnologici;
- fissare i vincoli di design.

2.3.3 Descrizione

Il processo di sviluppo si articola in:

- Analisi dei requisiti
- Progettazione
- Codifica

2.3.4 Attività

2.3.4.1 Analisi dei Requisiti

2.3.4.1.1 Scopo

Gli Analisti hanno il compito di redigere il documento di "Analisi dei Requisiti" che individua ed elenca dunque, i requisiti. Si suddividono in:

- definire lo scopo del lavoro;
- fornire ai progettisti riferimenti precisi ed affidabili; fissare le funzionalità e i requisiti concordati col cliente;
- fornire una base per raffinamenti successivi al fine di garantire un miglioramento continuo del prodotto e del processo di sviluppo;
- fornire ai verificatori riferimenti per l'attività di test circa i casi d'uso;
- stimare i costi.

2.3.4.1.2 Aspettative

Obiettivo dell'attività è la creazione della documentazione formale contenente tutti i requisiti richiesti dal proponente.

2.3.4.1.3 Descrizione

I requisiti si raccolgono secondo modalità predefinite:

- lettura del capitolato, analisi e approfondimento dello stesso;
- confronto con il proponente;
- confronto tra membri del team di progetto;
- possono emergere dall'analisi di uno o più casi d'uso.

2.3.4.1.4 Casi d'uso

Rappresenta un diagramma che esprime un comportamento, offerto o desiderato, sulla base di risultati osservabili. La struttura dei casi d'uso è così suddivisa:

- codice identificativo;
- titolo;
- diagramma UML;

- attori primari;
- attori secondari;
- descrizione;
- scenario;
- inclusioni(se presenti);
- esclusioni(se presenti);
- preconditione;
- postcondizione.

2.3.4.1.5 Codice identificativo dei casi d'uso

Il codice di ogni caso d'uso seguirà questo formalismo:

UC{codice_padre}.{codice_livello}

Dove:

- **codice_padre**: numero che identifica univocamente i casi d'uso;
- **codice_livello**: numero progressivo che identifica i sottocasi. Può a sua volta includere altri livelli.

2.3.4.1.6 Requisiti

Ogni requisito è composto dalla seguente struttura:

- codice identificativo;
- descrizione;
- fonti.

2.3.4.1.7 Codice identificativo dei requisiti

Ogni requisito è conforme alla seguente codifica:

R[Importanza][Tipologia][Codice]

Il significato delle cui voci è:

- **Importanza** distingue i requisiti in:
 - 1 Obbligatorio;
 - 2 Desiderabile;
 - 3 Opzionale.
- **Tipologia** distingue i requisiti in:
 - F Funzionale;
 - P Prestazionale;
 - Q Qualitativo;
 - V Vincolo.
- **Codice** è un identificatore univoco del requisito in forma gerarchica padre/figlio.

Ogni requisito contiene inoltre una descrizione testuale.

2.3.4.1.8 UML

I diagrammi UML devono essere realizzati usando la versione del linguaggio v2.0.

2.3.4.2 Progettazione

2.3.4.2.1 Scopo

L'attività di progettazione definisce, in funzione dei requisiti specificati nel documento "Analisi dei requisiti", le caratteristiche del prodotto software richiesto. Il compito di questa fase è di definire una soluzione del problema che sia soddisfacente per tutti gli stakeholders. La progettazione segue il procedimento inverso rispetto all'Analisi dei requisiti che divide il problema in parti per capirne completamente il dominio applicativo. La progettazione, infatti, rimette insieme le parti specificando le funzionalità dei sottosistemi in modo da ricondurre ad un'unica possibile soluzione.

2.3.4.2.2 Aspettative

Il processo ha come risultato la redazione dei documenti sopra citati. Ciò permetterà coerenza ed affidabilità in funzione del prodotto finale.

2.3.4.2.3 Descrizione

Le parti principali della Progettazione sono due:

- **specifica tecnica:** contiene le specifiche della progettazione ad alto livello del prodotto e delle sue componenti; vengono elencati i diagrammi UML che verranno utilizzati per la realizzazione dell'architettura e i test di verifica.
- **definizione di prodotto:** dettaglia ulteriormente l'attività di progettazione, integrando ciò che è riportato nella specifica tecnica. Definisce infine, i test necessari alla verifica.

2.3.4.2.4 Specifica tecnica

Redatta dal progettista, dovrà includere:

- **Diagrammi UML:**
 - Diagrammi delle classi;
 - Diagrammi dei package;
 - Diagrammi di attività;
 - Diagrammi di sequenza.
- **Tecnologie utilizzate:** devono essere descritte le tecnologie adottate specificandone l'utilizzo nel progetto, i vantaggi e gli svantaggi;
- **Design pattern:** devono essere descritti i design pattern utilizzati per realizzare l'architettura. Ogni design pattern deve essere accompagnato da una descrizione ed un diagramma, che ne esponga il significato e la struttura;
- **Tracciamento delle componenti:** ogni requisito deve riferirsi al componente che lo soddisfa.
- **Test di integrazione:** l'unione delle parti, intese come classi di verifica, permette di verificare che ogni componente del sistema funzioni nella maniera voluta.

2.3.4.2.5 Definizione di prodotto

A carico del progettista c'è anche la definizione di prodotto che si sofferma su diversi aspetti tra i quali:

- **Definizione delle classi:** ogni classe deve essere descritta in modo da spiegarne in maniera esaustiva lo scopo e le funzionalità, evitando ridondanze;
- **Tracciamento delle classi:** ogni requisito deve essere tracciato in modo da garantire che ogni classe ne soddisfi almeno uno e poter risalire alle classi a esso associate.
- **Test di unità:** devono essere definiti al fine di verificare che le parti funzionino individualmente nel modo stabilito.

2.3.4.3 Codifica

2.3.4.3.1 Scopo

Questa attività ha come scopo quello di normare l'effettiva realizzazione del prodotto software richiesto. In questa fase si concretizza la soluzione attraverso la programmazione. I programmatori dovranno attenersi a queste norme durante la fase di programmazione e implementazione.

2.3.4.3.2 Aspettative

L'uso di norme e convenzioni in questa fase, è fondamentale per permettere la generazione di codice leggibile e uniforme, agevolare le fasi di manutenzione, verifica e validazione e migliorare la qualità di prodotto.

2.3.4.3.3 Descrizione

La scrittura del codice dovrà rispettare quanto stabilito nella documentazione di prodotto e dovrà perseguire gli obiettivi di qualità definiti all'interno del documento "Piano di qualifica" per poter garantire una buona qualità al codice.

2.3.4.3.4 Stile di codifica

Al fine di garantire uniformità nel codice del progetto ciascun membro del gruppo è tenuto a rispettare le seguenti norme:

- **Indentazione:** I blocchi innestati devono essere correttamente indentati, usando per ciascun livello di indentazione quattro (4) spazi (fanno eccezione i commenti). Al fine di assicurare il rispetto di questa regola si consiglia di configurare adeguatamente il proprio editor o IDE;
- **Parentesizzazione:** è richiesto di inserire le parentesi di delimitazione dei costrutti in linea e non al di sotto di essi;
- **Scrittura dei metodi:** quando possibile mantenere i metodi brevi. Spesso però, metodi "lunghi" sono più appropriati, quindi questa regola è da considerare fra quelle consigliate;
- **Univocità dei nomi:** classi, metodi, variabili devono avere un nome univoco ed esplicativo al fine di evitare quanto più possibile ambiguità di comprensione.
- **Classi:** i nomi delle classi iniziano sempre con una lettera maiuscola;
- **Costanti:** i nomi delle costanti vanno scritte usando solo maiuscole;

- **Metodi:** i nomi dei metodi iniziano con una lettera minuscola; se sono composti da più parole le successive devono iniziare con una lettera maiuscola.
- **Lingua:** il codice, come anche i commenti deve essere scritto in lingua inglese;

2.3.4.3.5 Ricorsione

L'uso della ricorsione va evitato quanto più possibile in quanto potrebbe indurre ad una maggiore occupazione di memoria rispetto a soluzioni iterative.

2.3.5 Strumenti

Di seguito sono elencati gli strumenti utilizzati dal gruppo durante il progetto.

2.3.5.1 Draw.io

Per la produzione dei diagrammi UML viene utilizzato Draw.io in quanto offre molte agevolazioni per la produzione veloce dei diagrammi e risulta semplice da usare. <https://www.draw.io/>

2.3.5.2 IntelliJ IDEA

IntelliJ IDEA viene utilizzato per la codifica in Java e JavaScript. Questo IDE offre piena compatibilità con Linux, Windows, macOS, oltre ad essere un potente editor con molte funzionalità integrate. <https://www.jetbrains.com/idea/>

3 Processi di Supporto

3.1 Documentazione

3.1.1 Scopo

Ogni processo e attività significativi volti allo sviluppo del progetto sono documentati. Lo scopo di questa sezione è definire gli standard che riguardano i documenti prodotti durante il ciclo di vita del prodotto. I documenti sono consultabili nell'apposita sezione della repository.

3.1.2 Aspettative

Le aspettative su questo processo riguardano:

- un'idea precisa relativa alla struttura della documentazione che deve essere prodotta durante il ciclo di vita del software;
- l'individuazione di una serie di norme per la stesura di documenti coerenti e validi;

3.1.3 Descrizione

Questo capitolo contiene le decisioni e le norme che sono state scelte per la stesura, verifica e approvazione riguardante la documentazione ufficiale. Tali norme sono tassative per tutti i documenti formali.

3.1.4 Ciclo di vita del documento

Ogni documento segue le fasi del seguente ciclo di vita:

1. Creazione del documento: dopo attenta valutazione, nel momento in cui diventa necessario, il documento viene creato, applicando le norme per i documenti e adeguandosi a documenti precedenti dello stesso tipo; si usa un template se disponibile;
2. Creazione della struttura: se il documento è sufficientemente grande, viene creato il suo indice, che traccia gli argomenti trattati nel documento;
3. In lavorazione: il documento viene scritto in modo incrementale e per moduli;
4. In revisione: ogni voce del documento è stata prodotta interamente, ma è soggetta a revisioni per correggere, ampliare, semplificare quanto presente; se possibile, la revisione di ciascun frammento è svolta da almeno una persona diversa da chi ha scritto quel frammento;
5. Approvato: il responsabile di progetto sancisce che il documento è stato completato; esso è pronto per il rilascio.

Le fasi sono ordinate, ma non sono strettamente sequenziali. Sono consentiti ritorni a fasi precedenti.

3.1.5 Procedure

Per la stesura dei documenti il gruppo utilizza \LaTeX , un linguaggio di markup per la preparazione di testi, basato sul programma di composizione tipografica TEX.

3.1.5.1 Approvazione dei documenti

Quando si ritiene conclusa la stesura di un documento non formale, viene chiamato in causa il Responsabile di progetto. È sua responsabilità assegnare il documento ai Verificatori. Nel caso in cui i Verificatori individuino degli errori o delle difformità nel documento, sarà loro dovere informare il Responsabile di Progetto che rimetterà il documento nelle mani dei redattori. Il ciclo si ripete finché i verificatori non hanno più nulla da segnalare. Il documento verrà quindi passato al Responsabile di progetto, che si occuperà dell'approvazione o meno del documento. Nel caso decidesse di respingerlo, comunicherà ai redattori le modifiche da apportare. In casi estremi, potrà comunicare che la stesura del documento dovrà essere fatta ex-novo. Se il documento verrà approvato lo si riterrà un documento formale, e potrà essere distribuito alle persone nominate nella lista di distribuzione.

3.1.6 Template

Il gruppo ha creato un template \LaTeX per uniformare velocemente la struttura grafica e lo stile di formattazione dei documenti, in modo che i membri del team possano concentrarsi di più nella stesura del contenuto degli stessi. Lo scopo dei template è quello di permettere, a colui che redige il documento, di adottare automaticamente le conformità previste dal documento "Norme di Progetto". Permette inoltre di agevolare la procedura di adeguamento alle nuove norme per la redazione, nel caso esse cambino.

3.1.7 Struttura dei documenti

Un file `main.tex` (il cui "main" verrà sostituito dal nome del documento) raccoglie in input le sezioni di cui è composto il documento. Contiene un riferimento al file `config.tex`, contenente nuovi comandi \LaTeX che il gruppo utilizza per la stesura dei documenti, e uno al file `package.tex` che contiene tutti i package utili al linguaggio di markup.

3.1.7.1 Prima pagina

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

- **Logo del gruppo:** logo di 8Lab Solutions visibile come primo elemento centrato orizzontalmente in alto;
- **Gruppo e progetto:** nome del gruppo e del progetto "Soldino", visibile centralmente subito sotto il logo;
- **Titolo:** nome del documento, posizionato centralmente in grassetto;
- **Tabella descrittiva:** presente sotto il titolo del documento, centrale e contenente le seguenti informazioni:
 - "Versione" del documento;
 - "Approvazione": nome e cognome dei membri del gruppo incaricati dell'approvazione del documento;
 - "Redazione": nome e cognome dei membri del gruppo incaricati della redazione del documento;
 - "Verifica": nome e cognome dei membri del gruppo incaricati della verifica del documento;
 - "Stato" del documento: lo stadio corrente del ciclo di vita del documento;

- "Uso": tipo d'uso che può essere o interno o esterno;
- "Destinato a": destinatari del documento.
- **Descrizione:** descrizione sintetica relativa al documento, centrale, posta sotto la tabella descrittiva;
- **Recapito:** indirizzo di posta elettronica del gruppo, posizionato centralmente in fondo alla pagina.

3.1.7.2 Registro modifiche

Ogni documento, fatta eccezione per i verbali, deve disporre di un registro delle modifiche (changelog) a seguito della prima pagina. Tale registro contiene le modifiche apportate al documento stesso, e vengono indicati:

- versione del documento dopo la modifica;
- data in cui la modifica è stata apportata;
- nominativo di colui/colei che ha apportato la modifica al documento;
- ruolo ricoperto dalla persona che ha effettuato la modifica;
- descrizione sintetica della modifica.

3.1.7.3 Indice

Gli indici hanno lo scopo di riepilogare e dare una visione macroscopica della struttura del documento. Permettono quindi di rintracciare i contenuti tramite una gerarchia. Tale gerarchia è basata sul livello delle sezioni. Ogni documento, esclusi i verbali, dovrà essere corredato dall'indice dei contenuti. Esso sarà posizionato dopo il diario delle modifiche. Se sono presenti tabelle o immagini all'interno del documento, l'indice dei contenuti sarà seguito dalla lista delle tabelle e poi dalla lista delle figure.

3.1.7.4 Contenuto principale

Il resto del documento è occupato da pagine di contenuto che dovranno essere così strutturate:

- in alto a sinistra sarà presente il logo del gruppo lo stesso che è riportato nel frontespizio;
- in alto a destra dovrà essere riportato il nome del documento;
- una riga dovrà dividere l'intestazione dal contenuto;
- il contenuto della pagina sarà posto tra l'intestazione e il piè pagina;
- una riga dovrà dividere il contenuto dal piè pagina;
- in basso a destra sarà posto il numero della pagina corrente.

3.1.7.5 Note a piè pagina

In caso di presenza di note da spiegare, esse vanno indicate nella pagina corrente, in basso a sinistra. Ogni nota deve riportare un numero e una descrizione.

3.1.8 Versionamento

3.1.8.1 Processo di versionamento

Per le parti versionabili del progetto e documenti ufficiali si è scelto l'utilizzo della tecnologia Git, usando il servizio di GitHub. La condivisione dei documenti informali e delle parti non versionabili è invece effettuata tramite l'uso di una cartella su Google Drive.

3.1.8.2 Codice per la versione

Ogni documento, verbali esclusi, deve avere una storia, ricostruibile attraverso le sue versioni. Ogni versione deve corrispondere a una riga del registro delle modifiche. Ogni numero di versione è composto da tre cifre:

X.Y.Z

- **X**: rappresenta una versione stabile del documento, resa tale dopo l' approvazione del Responsabile di progetto;
 - inizia da 0;
 - viene incrementato dal Responsabile di Progetto all'approvazione del documento;
- **Y**: rappresenta una versione parzialmente stabile del documento che è stata soggetta a verifica da parte di un Verificatore;
 - inizia da 0;
 - viene incrementato dal Verificatore ad ogni verifica;
 - quando viene incrementato X, viene riportato a 0.
- **Z**: rappresenta una versione instabile del documento in fase di lavorazione da parte dei redattori.
 - inizia da 0;
 - viene incrementato dal redattore del documento ad ogni modifica;
 - quando viene incrementato Y, viene riportato a 0.

3.1.9 Norme tipografiche

3.1.9.1 Convenzioni sui nomi dei file I nomi di file (escludendo l'estensione) e cartelle utilizzano la convenzione **snake case** e alcune regole aggiuntive elencate a seguito:

1. i nomi dei file composti da più parole usano underscore come carattere separatore;
2. i nomi sono scritti interamente in minuscolo;
3. le preposizioni non si omettono.

Alcuni esempi **corretti** sono:

- studio_di_fattibilità;
- analisi_dei_requisiti.

Alcuni esempi **non corretti** sono:

- Norme_di_progetto (usa maiuscole);
- norme-di-progetto (carattere separatore errato);
- norme_progetto (omette 'di').

3.1.9.2 Stile del testo

- **Glossario:** ogni voce del glossario sarà marcata con una *G* maiuscola a pedice; se la voce è presente in più occorrenze in un documento, è sufficiente che la prima occorrenza sia marcata;
- **Grassetto:** viene applicato ai titoli, agli elementi di un elenco puntato o a termini a cui si vuole far risaltare il significato all'interno delle frasi;
- **Maiuscolo:** vengono scritti con sole lettere maiuscole tutti gli acronimi.

3.1.9.3 Elenchi puntati

Ogni voce di un elenco comincia per lettera minuscola e termina per ";" eccetto l'ultima che termina per ".". I sottoelenchi sono innestati dentro una voce di elenco e rispettano le medesime regole, poiché la loro funzione è analoga.

3.1.9.4 Formati comuni

In conformità allo standard ISO 8601, le date devono essere scritte secondo il formato:

YYYY-MM-DD

- **YYYY:** rappresentazione dell'anno con quattro (4) cifre;
- **MM:** rappresentazione del mese con due (2) cifre;
- **DD:** rappresentazione del giorno con due (2) cifre;

3.1.9.5 Sigle

Il progetto prevede la redazione di un insieme di documenti, suddivisi in documenti interni e documenti esterni. Essi sono elencati di seguito con le rispettive sigle.

I documenti esterni sono:

- **AdR:** Analisi dei Requisiti: stabilisce le caratteristiche che il software deve rispettare;
- **MU:** Manuale Utente: ad uso degli utilizzatori del software;
- **MS:** Manuale Sviluppatore: per gli sviluppatori e manutentori;
- **PdP:** Piano di Progetto: concerne la gestione del progetto, evidenziandone la fattibilità e le criticità; tratta di tempi, costi, obiettivi, rischi, vincoli;
- **PdQ:** Piano di Qualifica: descrive la qualità del software e dei processi, e come la si intende raggiungere mediante l'uso di strumenti, metriche e processi;
- **RA:** Revisione di Accettazione: determina l'entrata;
- **RP:** Revisione di Progettazione;
- **RQ:** Revisione di Qualifica;
- **RR:** Revisione dei Requisiti.

I documenti interni sono:

- **G:** Glossario: raccoglie i termini di interesse per il team di sviluppo e sui quali il team vuole concordare;

- **NdP**: Norme di Progetto: sono un riferimento per lo svolgimento delle attività di progetto;
- **SdF**: Studio di Fattibilità: descrive sommariamente i capitolati e spiega la loro scelta o esclusione;
- **V**: Verbale: descrizione le interazioni e i loro prodotti avvenuti durante un incontro con il proponente o un incontro interno di interesse per il progetto; sono orientati a dare informazioni semplici e immediatamente fruibili.

3.1.10 Elementi grafici

3.1.10.1 Changelog I changelog sono tabelle delle modifiche da inserire nella maggioranza dei documenti richiesti (eccetti alcuni, come i verbali. Contengono i seguenti titoli nell'intestazione:

- versione;
- data;
- nominativo;
- ruolo;
- descrizione.

Ogni voce in tabella corrisponde a una modifica significativa del documento: un incremento, una revisione o un'approvazione.

3.1.10.2 Tabelle

Ogni tabella deve essere accompagnata dalla propria didascalia e una breve descrizione; in tale didascalia deve comparire il numero della sezione a cui si riferisce, seguita in modo incrementale dal numero progressivo delle tabelle di quella sezione.

- **X.Y**: rappresenta la sezione;
- **Z**: rappresenta il numero progressivo della tabella nella sezione;

Fanno eccezione le tabelle del registro delle modifiche (changelog) che non hanno nessuna descrizione e le tabelle dei casi d'uso presenti nel documento "Analisi dei requisiti".

3.1.10.3 Immagini

Le immagini seguono lo stesso comportamento delle tabelle per quanto riguarda didascalia e descrizione. Tutti i diagrammi UML vengono inseriti nei documenti sotto forma di immagine.

3.1.11 Classificazione dei documenti

3.1.11.1 Documenti informali

Tutti i documenti saranno da ritenersi informali fino all'approvazione da parte del Responsabile di progetto. Egli potrà eventualmente richiederne una revisione. L'uso dei documenti informali è da considerarsi esclusivamente interno al team. Tale utilizzo è circoscritto alla sola redazione di tali documenti.

3.1.11.2 Documenti formali

Un documento viene definito formale quando è stato approvato dal Responsabile di Progetto. Solo i documenti formali possono essere forniti ai soggetti presenti nella lista di distribuzione. Per arrivare a tale stato il documento deve aver già superato la verifica e la validazione. Ogni volta che un documento formale verrà modificato, la nuova versione è da considerarsi non formale. Rimarrà tale fino alla sua successiva approvazione da parte del Responsabile di progetto. Sarà quindi trattata al pari di un documento informale.

3.1.11.3 Verball

I verball vengono prodotti dal soggetto incaricato alla loro stesura in occasione di incontri tra i membri del team e/o gli esterni. Per essi è prevista un'unica stesura. Tale scelta è motivata dal fatto che apportare modifiche implica una modifica delle decisioni prese in modo retroattivo. I verball si compongono della seguente struttura:

- **Luogo:** luogo di svolgimento dell'incontro;
- **Data:** data dell'incontro(formato YYYY-MM-DD);
- **Ora di inizio:** l'orario di inizio dell'incontro;
- **Ora di fine:** l'orario di fine dell'incontro;
- **Partecipanti del gruppo:** i nominativi dei membri che erano presenti all'incontro;
- **Partecipanti esterni** (se presenti): i nominativi di persone esterne al gruppo che hanno partecipato all'incontro.
- **Argomenti affrontati:** ciò di cui si è discusso durante l'incontro. E' composto da una "Descrizione" e da un "Riepilogo tracciamenti".

Ogni verball dovrà essere denominato secondo il seguente formato:

TipologiaYYYY-MM-DD

dove per "Tipologia" si intende il tipo di verball:

- **Interno:** concentrato sul riassunto dell'incontro dei membri del team;
- **Esterno:** concentrato sulla trattazione di argomenti con partecipanti esterni al gruppo, in particolare domande e risposte riguardanti il progetto in sè.

La sezione "Riepilogo tracciamenti" avrà la funzione di tenere traccia delle decisioni emerse da ogni incontro, sotto forma di tabella riassuntiva a fine di ogni verball. Il formalismo utilizzato dalla tabella dovrà essere il seguente:

VTipologia_X.Y

Dove:

- per "Tipologia" basterà indicare l'iniziale della tipologia del verball(I=Interno/E=Esterno).
- "X.Y" si intende il numero progressivo della decisione presa dal gruppo(partendo da 1).

3.1.12 Strumenti

3.1.12.1 L^AT_EX

Lo strumento scelto per la scrittura di documenti è L^AT_EX. La sua utilità sta nel permettere di scrivere documenti in modo ordinato, modulare, collaborativo e scalabile.

3.1.12.2 TexStudio

Per la stesura del codice \LaTeX è stato utilizzato l'editor TexStudio. Questo strumento include oltre a integrare un compilatore e visualizzatore PDF, fornisce suggerimenti di completamento per comandi \LaTeX .

<https://www.texstudio.org/>

3.1.12.3 Draw.io

Draw.io è una piattaforma online per la produzione di varie tipologie di diagrammi, tra cui gli UML.

<https://www.draw.io/>

3.2 Qualità

Per la trattazione della qualità si fa riferimento al Piano di Qualifica, dove sono descritte le modalità utilizzate per garantire la qualità nello sviluppo del progetto. In particolare, in tale documento:

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

Per ogni processo si mostrano:

- gli obiettivi da perseguire;
- le strategie da applicare;
- le metriche da utilizzare.

al fine di migliorare e controllare i processi e i loro risultati. Per ogni prodotto si mostrano:

- gli obiettivi da perseguire;
- le metriche da utilizzare;

al fine di ottenere software e documentazione di sufficiente qualità.

3.3 Verifica

3.3.1 Scopo

Il processo di verifica ha per scopo l'ottenimento di prodotti corretti, coesi e completi. Gli oggetti della verifica sono il software e i documenti prodotti.

3.3.2 Aspettative

Il corretto svolgimento del processo di verifica rispetta i punti seguenti:

- La verifica è effettuata seguendo procedure definite;
- Vi sono criteri chiari e affidabili da perseguire per verificare;
- I prodotti passano attraverso fasi successive, ognuna delle quali è verificata;
- Dopo la verifica il prodotto è in uno stato stabile;
- Il processo di validazione, che è successivo alla verifica, diviene ben fondato e più semplice;

3.3.3 Descrizione

Il processo di verifica prende in input ciò che è già stato prodotto, e lo restituisce in uno stato conforme alle aspettative. Per ottenere tale risultato ci si affida a processi di analisi e di test.

3.3.4 Attività

3.3.4.1 Analisi

Il processo di analisi si suddivide in analisi statica e analisi dinamica.

3.3.4.1.1 Analisi statica

L'analisi statica effettua controlli su: documenti e codice. Di essi valuta e applica la correttezza (intesa come assenza di errori e difetti), la conformità a regole e la coesione dei componenti. Per effettuare analisi statica esistono metodi manuali di lettura (attuati da persone) e metodi formali (attuati da macchine). I metodi manuali sono due:

- **Walkthrough:** i vari componenti del team analizzano gli oggetti nella loro totalità per cercare difetti ad ampio spettro, non sapendo dove questi possano essere;
- **Inspection:** i verificatori fanno ispezione sugli oggetti in cui si cercano difetti specifici, noti prima dell'ispezione e raggruppati in checklist;

A seguire sono descritte le liste di controllo per le ispezioni.

Difetto	Descrizione
Errore di conformità	Formato di documento, data o nome errato
Errore di sintassi	Sintassi scorretta o poco leggibile
Errore di contenuto	Contenuto errato
Parte mancante	Manca una parte. Controllare titoli vuoti e PLACEHOLDER

Tabella 2: elenca gli errori più frequenti nei documenti

Errore	Descrizione
Errore di nomenclatura	Il nome non rispetta la rispettiva norma o non è dichiarativo della propria funzione
Eccesso di responsabilità	La classe o il metodo può essere suddiviso in parti più piccole

Tabella 3: elenca gli errori più frequenti nel codice

3.3.4.1.2 Analisi dinamica

L'analisi dinamica è una tecnica di analisi del prodotto software che richiede la sua esecuzione. Viene effettuata mediante dei test volti a verificare il funzionamento del prodotto e nel caso in cui vengano riscontrate anomalie ne permette l'identificazione. I test devono essere ripetibili, cioè deve essere possibile, dato lo stesso input e nello stesso ambiente, risalire allo stesso output. Per ogni test devono dunque essere definiti i seguenti parametri:

- **Ambiente:** sistema hardware e software sul quale verrà eseguito il test del prodotto;
- **Stato iniziale:** lo stato iniziale dal quale il test viene eseguito;
- **Input:** input inserito;

- **Output:** output atteso;
- **Istruzioni aggiuntive:** ulteriori istruzioni su come va eseguito il test e su come vanno interpretati i risultati ottenuti.

3.3.4.2 Test

I test sono l'attività fondamentale dell'analisi dinamica: verificano che il codice scritto funzioni correttamente. Test ben scritti devono:

- Essere ripetibili;
- Specificare l'ambiente di esecuzione;
- Specificare input, output richiesti;
- Avvertire di possibili side-effect;
- Fornire informazioni sui risultati dell'esecuzione per una futura analisi.

Tale verifica è eseguita a vari livelli all'interno del software.

3.3.4.2.1 Test di unità

I test di unità si eseguono su unità di software. Un unità di software è atomica, coesa e viene scritta da un singolo individuo: quindi, per costruzione, tende ad essere ben leggibile e di facile verifica. Le singole unità possono essere testate con l'ausilio di driver e stub.

3.3.4.2.2 Test di integrazione

Dopo aver superato i test di unità, le unità vengono assemblate in gruppi progressivamente più grandi per testare le relazioni tra le parti. Un agglomerato che supera il test di integrazione costituisce quindi una nuova unità per un agglomerato di grandezza maggiore. Questa procedura si ripete fino a raggiungere la grandezza totale del sistema.

3.3.4.2.3 Test di sistema

Una volta raggiunta la grandezza del sistema, esso viene testato nella sua interezza. In questa fase ci si assicura che il sistema rispetti tutte le specifiche definite nell'*AdR*.

3.3.4.2.4 Test di regressione

Si effettua di seguito a una modifica del sistema, e consiste nella riesecuzione dei test esistenti.

3.3.4.2.5 Test di collaudo

Simile al test di sistema, ma eseguito con la collaborazione dei committenti, si occupa di verificare il prodotto intero, e in particolare la sua conformità alle aspettative. Il superamento del test di collaudo dichiara che il software è pronto per essere rilasciato.

3.3.5 Strumenti

3.3.5.1 Verifica ortografica

Viene utilizzata la verifica dell'ortografia in tempo reale, strumento integrato in TexStudio che sottolinea in rosso le parole errate secondo la lingua italiana.

3.3.5.2 Validazione W3C

Per la validazione delle pagine di markup HTML viene utilizzato lo strumento offerto dal W3C, raggiungibile al seguente indirizzo:

<https://validator.w3.org/>

Per la validazione dei fogli di stile CSS viene utilizzato lo strumento offerto dal W3C, raggiungibile al seguente indirizzo:

<https://jigsaw.w3.org/css-validator/>

4 Processi Organizzativi

4.1 Gestione Organizzativa

4.1.1 Scopo

Il processo di gestione contiene le attività di carattere gestionale. Queste attività consistono principalmente nell'istanziare processi nel progetto, pianificare e stimare tempi e costi, assegnare ruoli e rispettivi compiti ai membri del team, nonché controllare l'esecuzione e i risultati di tutti i processi in atto. La gestione compete al Responsabile di Progetto, che ha il compito di gestire il progetto, i suoi processi, i suoi prodotti e i membri del team.

4.1.2 Aspettative

Gli obiettivi di questo processo sono:

- Ottenere una pianificazione ragionevole delle attività da seguire;
- Coordinare i membri del team assegnando loro ruoli, compiti e facilitando la comunicazione tra loro;
- Adoperare processi per regolare le attività e renderle economiche;
- Mantenere il controllo sul progetto in modo efficace ma non invasivo, monitorando il team, i processi e i prodotti.

4.1.3 Descrizione

Le attività di gestione seguono il flusso di fasi seguente:

- inizio e definizione dello scopo;
- pianificazione;
- esecuzione e controllo;
- revisione e valutazione;
- chiusura.

4.1.4 Ruoli di progetto

Ciascun membro del gruppo, a rotazione, si impegnerà a ricoprire i ruoli che corrispondono alle omonime figure aziendali. Nel Piano di Progetto vengono organizzate e pianificate le attività assegnate ai specifici ruoli previsti nell'attività di progetto. I ruoli che ogni componente del gruppo è tenuto a rispettare sono descritti di seguito.

4.1.4.1 Responsabile di progetto

Il responsabile di progetto è una figura importante in quanto ricadono su di lui le responsabilità di pianificazione, gestione, controllo e coordinamento. Un altro ruolo del responsabile è quello di rappresentare l'interfaccia del gruppo con le persone esterne ad esso: sono quindi di sua competenza le comunicazioni con committente e proponente.

Il responsabile deve quindi:

- gestire, controllare e coordinare risorse e attività del gruppo;
- gestire, controllare e coordinare gli altri componenti del gruppo;
- analizzare e gestire le criticità;
- approvare i documenti.

4.1.4.2 Amministratore di progetto

L'amministratore ha il compito di supporto e controllo dell'ambiente di lavoro.

L'amministratore deve quindi:

- amministrazione delle infrastrutture di supporto;
- risoluzione dei problemi legati alla gestione dei processi;
- gestione della documentazione;
- controllo di versioni e configurazioni.

4.1.4.3 Analista

L'analista si occupa di analisi dei problemi e del dominio applicativo. Questa figura non sarà sempre presente durante il progetto.

L'analista deve quindi:

- studio del dominio del problema e del problema stesso, definendone complessità e requisiti;
- redazione di Analisi dei Requisiti e Studio di Fattibilità.

4.1.4.4 Progettista

Il progettista gestisce gli aspetti tecnologici e tecnici del progetto.

Il progettista deve:

- effettuare scelte efficienti ed ottimizzate su aspetti tecnici del progetto;
- sviluppare un'architettura che sfrutti tecnologie note ed ottimizzate su cui basare un prodotto stabile e mantenibile.

4.1.4.5 Programmatore

Il programmatore è il responsabile della codifica del progetto e delle componenti di supporto, che serviranno per effettuare le prove di verifica e validazione sul prodotto.

Il programmatore deve:

- implementare le decisioni del progettista;
- creare o gestire componenti di supporto per la verifica e la validazione del codice.

4.1.4.6 Verificatore

Il verificatore si occupa di controllare il prodotto del lavoro svolto dagli altri membri del team, sia esso codice o documentazione. Per le correzioni si affida agli standard definiti nelle Norme di Progetto, nonché alla propria esperienza e capacità di giudizio.

Il verificatore deve:

- ispezionare i prodotti in fase di revisione, avvalendosi delle tecniche e degli strumenti definiti nelle NdP;
- correggere difetti ed errori del prodotto in esame;
- decidere se applicare le correzioni in autonomia o in collaborazione con l'autore del prodotto in questione, a seconda della natura dell'errore.

4.1.5 Procedure

Seguono le modalità che il gruppo adotterà durante la realizzazione del progetto. Le comunicazioni saranno interne, ossia coinvolgono i partecipanti del gruppo, o esterne, ossia coinvolgono anche proponente e committente.

4.1.5.1 Gestione delle comunicazioni

4.1.5.1.1 Comunicazioni interne

Le comunicazioni interne al gruppo avvengono utilizzando Slack. Slack è un software di collaborazione aziendale adatto al lavoro di gruppo: permette di separare i vari argomenti di discussione in canali e i team in workspace; permette anche di integrare servizi di terze parti utili al team, tra cui bot e chiamate. Le sue funzionalità e la sua conformità agli scopi del progetto ne hanno favorito la scelta.

4.1.5.1.2 Comunicazioni esterne

Le comunicazioni con soggetti esterni al gruppo sono di competenza del responsabile. Gli strumenti predefiniti sono la posta elettronica, che utilizza l'indirizzo 8labsolutions@gmail.com creata in fase di creazione del team e, su suggerimento del proponente, un canale Slack creato proprio da Red Babel. Il responsabile ha il compito di tenere informati gli altri componenti del gruppo.

4.1.5.2 Gestione degli incontri

4.1.5.2.1 Incontri interni del team

Le riunioni interne del team sono organizzate dal responsabile in accordo con i membri del team. Per stabilire data e ora viene utilizzato Google Calendar, dove ogni membro segnala gli orari nel quale ha impegni.

4.1.5.2.2 Verbali di riunioni interne

Ad ogni riunione interna fatta corrisponderà un verbale. Questo sarà redatto da segretario, persona nominata dal responsabile, che dovrà tenere nota delle discussioni fatte e delle decisioni prese.

4.1.5.2.3 Incontri esterni del team

Il responsabile ha il compito di comunicare e organizzare gli incontri con proponente e committente. Se un membro del gruppo, il proponente o il committente ritiene necessario organizzare un incontro il responsabile decide una data, in accordo tra le due parti, e la comunica tramite i canali sopra citati.

Le riunioni esterne sono di grande importanza per la buona riuscita del progetto: a esse si presta molta cura.

4.1.5.2.4 Verbali esterni di riunione

Come per le riunioni interne, anche per le esterne viene redatto un verbale. La struttura dei due tipologie è analoga, ma essendo le riunioni esterne di maggiore criticità, il verbale va redatto con maggiore attenzione, evitando di tralasciare informazioni importanti.

4.1.5.3 Gestione degli strumenti di coordinamento

4.1.5.3.1 Ticketing

Il ticketing consente ai membri di avere chiaro in ogni momento quali attività sono in corso; permette al RdP di assegnare compiti ai membri del team e di controllare l'andamento delle attività; permette ai membri del team di conoscere e gestire il proprio carico di lavoro.

Lo strumento di ticketing scelto è Trello: consiste in una lavagna virtuale online dove ogni task corrisponde a un compito, a cui sono associati una data di scadenza e un insieme di membri assegnatari. Ogni compito passa attraverso i seguenti stati:

- da fare;
- in lavorazione;
- in revisione;
- completato.

La scelta di Trello è stata determinata dalla sua facilità di apprendimento, di controllo, dall'usabilità e dalla trasparenza che fornisce. La gestione dei ticket dev'essere scrupolosa, perché la lavagna di Trello tende ad affollarsi velocemente.

4.1.5.4 Gestione degli strumenti di versionamento

4.1.5.4.1 Repository

Il sistema di versionamento usato è *Git*, noto sistema di controllo versione distribuito. I membri del team possono usare lo strumento sia da linea di comando sia attraverso software che ne migliorano l'usabilità, come GitFlow, GitKraken. La versione comune e ufficiale del progetto è ospitata in una repository remota su Github all'indirizzo <https://github.com/8LabSolutions/Soldino>. Questa repository remota è pubblica quindi accessibile a tutti, inclusi cliente e committente.

4.1.5.4.2 Struttura del repository

Ci sono due tipi di repository:

- remota: ospitata su *GitHub*, è la versione di riferimento e quella disponibile all'esterno del team;

- locale: ogni studente possiede una copia locale del progetto su cui lavorare;

Entrambe i tipi di repository sono organizzati in cartelle:

- latex: qui risiedono i file di carattere generale, come i file relativi a package, che vengono inseriti in tutti i documenti, e i template, che sono usati come riferimento per nuovi documenti di struttura già definita;
- guide: qui si trovano delle brevi guide di carattere pratico, come brevi spiegazioni per applicare comandi latex poco conosciuti;
- *RR*: qui si trovano i documenti, suddivisi tra esterni ed interni, la cui consegna è dovuta alla Revisione dei Requisiti;
- Saranno aggiunte in futuro cartelle distinte nominate **RP**, **RQ** e **RA** contenenti i file delle rispettive consegne.

Lo scopo della suddivisione dei file per revisioni è di evidenziare il lavoro svolto per ogni milestone maggiore, e di rendere ben tracciabili i file che le riguardano.

4.1.5.4.3 Tipi di file e .gitignore

I file di interesse per il progetto sono:

- file .tex di L^AT_EX;
- file .pdf che sono oggetto di consegna;
- alcuni file testuali e immagini di supporto ai precedenti.

Il file .gitignore presente al livello più esterno della repository elenca ed esclude tutti i file che vengono prodotti durante lo sviluppo del progetto ma non sono rilevanti per la consegna.

4.1.5.4.4 Lavoro collaborativo e versionamento

La repository *git* è composta di vari rami per organizzare il lavoro e renderlo modulare e collaborativo. Ogni membro del team lavora in locale su un branch della repository; si può lavorare in più su uno stesso ramo; è auspicabile lavorare parallelamente su rami differenti a diverse attività parallele.

La sequenza di attività che un membro del team usa produrre un incremento da aggiungere al progetto è la seguente:

1. scelta del branch corretto;
2. pull dalla repository remota per aggiornare il proprio branch locale;
3. svolgimento del lavoro, che risulta nella modifica dei propri file locali;
4. aggiunta dei file al sistema di versionamento;
5. commit dei file aggiunti, corredato di un messaggio che identifica le modifiche;
6. push del commit nella repository remota;

La routine serve a massimizzare la consistenza e minimizzare i conflitti tra le repository locali e quelle remote, così da rendere il lavoro il più possibile piacevole e snello.

4.1.5.5 Gestione dei rischi

Il Responsabile di Progetto ha il compito di rilevare i rischi e mantenerli, documentando quest'attività nel Piano di Progetto. La procedura da seguire per la gestione dei rischi è la seguente:

- individuare nuovi problemi e monitorare i rischi già previsti;
- registrare ogni riscontro previsto dei rischi nel Piano di Progetto;
- aggiungere i nuovi rischi individuati nel Piano di Progetto;
- ridefinire, se necessario, le strategie di progetto.

4.1.5.5.1 Codifica dei rischi Le tipologie di rischi sono così codificate:

- **RT:** Rischi Tecnologici;
- **RO:** Rischi Organizzativi;
- **RI:** Rischi Interpersonali.

4.1.5.6 Gestione di progetto Per dividere le attività da svolgere e per assicurarsi che tutto venga svolto entro una deadline si è scelto di utilizzare Trello. Questo permette di assegnare una specifica attività ad un membro e impostare anche una scadenza entro la quale l'attività dovrà essere terminata. Sarà il responsabile a gestire la suddivisione degli incarichi tra i membri e a determinare le scadenze per queste.

4.1.6 Strumenti utilizzati

Il gruppo, nel corso del progetto, ha utilizzato o utilizzerà i seguenti strumenti:

- **Telegram:** strumento di messaggistica utilizzato per un'iniziale gestione del gruppo;
- **Slack:** per la comunicazione interna del team ed eventuali comunicazioni col proponente;
- **Trello:** per assegnare determinate attività e imporre una scadenza;
- **Git:** sistema di controllo di versionamento;
- **Gitflow, GitKraken:** interfacce per utilizzare Git più comodamente sul proprio desktop;
- **GitHub:** per il versionamento e il salvataggio in remoto di tutti i file riguardanti il progetto.
- **Google Drive e Documenti:** utilizzato per la stesura di file che sono soggetti a molti cambiamenti e devono essere visibili a tutti nella sua versione più aggiornata come ad esempio il glossario;
- **Google Calendar:** per facilitare il lavoro al responsabile ogni settimana ciascun membro indica quando non è disponibile per un incontro;
- **Skype:** servizio che offre possibilità di fare video conferenze e chiamate VoIP, utilizzato per il primo contatto con il proponente;
- **Hangouts:** servizio che offre la possibilità di fare video conferenze e chiamate VoIP, utilizzato per parlare con il proponente e alcuni incontri interni;
- **Sistemi operativi:** i requisiti non indicano la necessità di usare un sistema operativo specifico, verranno quindi utilizzati sia Windows, sia Linux, sia Mac OS dai diversi membri del team;

4.2 Formazione del team

Per quando riguarda l'apprendimento basilare delle tecnologie richieste per il progetto, il gruppo ha optato per l'auto-formazione. Le conoscenze acquisite sono condivise in maniera informale e consolidate o migliorate; le conoscenze più critiche sono condivise anche in forma scritta: nella repository si trovano infatti una Wiki per gli argomenti più comune e la cartella guide per alcune utilità pratiche minori. In caso di membri del team in difficoltà, i membri più esperti offrono spiegazioni, riferimenti o chi ne ha bisogno.