

# AtAVi

Accoglienza tramite assistente virtuale

---

UNIVERSITÀ DI PADOVA

## NORME DI PROGETTO V4.00

~/anSwEr



[answer.group17@gmail.com](mailto:answer.group17@gmail.com)

## Informazioni sul documento

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>Nome Documento:</b>       | <i>NormeDiProgetto_v.4.00</i>     |
| <b>Versione:</b>             | 4.00                              |
| <b>Data di Creazione:</b>    | 1 Dicembre 2016                   |
| <b>Data ultima Modifica:</b> | 11 Maggio 2017                    |
| <b>Stato:</b>                | <b>Approvato</b>                  |
| <b>Redazione:</b>            | Leonardo Zoso                     |
| <b>Verifica:</b>             | Alessandro Bari,<br>Matteo Maran. |
| <b>Approvazione:</b>         | Leonardo Zoso                     |
| <b>Uso:</b>                  | Interno                           |
| <b>Distribuzione:</b>        | <i>Answer</i>                     |
| <b>Destinato a:</b>          | Prof. Vardanega, Prof. Cardin     |
| <b>Email di riferimento:</b> | answer.group17@gmail.com          |

## Sommario

Questo documento contiene le norme di progetto che il gruppo *Answer* seguirà durante la realizzazione del prodotto *AtAVi*.

## Registro delle modifiche

| Versione | Data       | Autore          | Ruolo                           | Descrizione  |
|----------|------------|-----------------|---------------------------------|--|
| 4.00     | 2017-05-12 | Leonardo Zoso   | <i>Responsabile di progetto</i> | Approvazione del documento.<br>Causa consegna del documento.                 |
| 3.09     | 2017-05-11 | Matteo Maran    | <i>Verificatore</i>             | Verifica del documento.<br>Causa modifiche effettuate.                       |
| 3.08     | 2017-05-11 | Leonardo Zoso   | <i>Amministratore</i>           | Aggiunti riferimenti ipertestuali.<br>Causa incremento documento.            |
| 3.07     | 2017-04-28 | Alessandro Bari | <i>Verificatore</i>             | Verifica del documento.<br>Causa modifiche effettuate.                       |
| 3.06     | 2017-04-27 | Leonardo Zoso   | <i>Amministratore</i>           | Aggiunta §2.1.2 Collaudo. Causa resoconto consegna.                          |
| 3.05     | 2017-04-27 | Leonardo Zoso   | <i>Amministratore</i>           | Aggiunta §2.2.2.4 Tecniche di modularizzazione.<br>Causa resoconto consegna. |
| 3.04     | 2017-04-27 | Leonardo Zoso   | <i>Amministratore</i>           | Aggiunta §2.2.2.3 Obiettivi. Causa resoconto consegna.                       |
| 3.03     | 2017-04-26 | Leonardo Zoso   | <i>Amministratore</i>           | Aggiunta §2.2.2.2 Requisiti per i progettisti.<br>Causa resoconto consegna.  |

| Versione | Data       | Autore            | Ruolo                           | Descrizione  |
|----------|------------|-------------------|---------------------------------|--|
| 3.02     | 2017-04-25 | Leonardo Zoso     | <i>Amministratore</i>           | Approfondita §2.2.2.1 Diagrammi UML. Causa incremento documento.                             |
| 3.01     | 2017-04-24 | Leonardo Zoso     | <i>Amministratore</i>           | Modificata struttura sezione §4 Processi organizzativi. Causa resoconto consegna.            |
| 3.00     | 2017-04-10 | Andrei Tabacariu  | <i>Responsabile di progetto</i> | Approvazione del documento. Causa consegna del documento.                                    |
| 2.04     | 2017-03-24 | Alessandro Bari   | <i>Verificatore</i>             | Verifica del documento. Causa modifiche effettuate.  |
| 2.03     | 2017-03-22 | Francesca Del Nin | <i>Amministratore</i>           | Spostata sezione test in §3.4.5 in quanto attinenti alla verifica. Causa resoconto consegna. |
| 2.02     | 2017-03-21 | Eugen Saraci      | <i>Amministratore</i>           | Aggiunta sezione §2.2.3.4.6 Formattazione. Causa incremento del documento.                   |
| 2.01     | 2017-03-20 | Francesca Del Nin | <i>Amministratore</i>           | Aggiunta sezione §4.3.4 Meccanismi di controllo. Causa incremento del documento.             |

| Versione | Data       | Autore            | Ruolo                           | Descrizione   |
|----------|------------|-------------------|---------------------------------|---|
| 2.00     | 2017-03-06 | Matteo Maran      | <i>Responsabile di progetto</i> | Approvazione del documento.<br>Causa consegna del documento.                              |
| 1.08     | 2017-02-27 | Eugen Saraci      | <i>Amministratore</i>           | Verifica del documento.<br>Causa modifiche effettuate.                                    |
| 1.07     | 2017-02-22 | Andrei Tabacariu  | <i>Amministratore</i>           | Aggiunta §2.2.3 Codifica. Causa incremento del documento.                                 |
| 1.06     | 2017-02-21 | Leonardo Zoso     | <i>Amministratore</i>           | Aggiunta §2.2.2.7 Design Pattern, §2.2.2.1 Diagrammi UML. Causa incremento del documento. |
| 1.05     | 2017-02-07 | Matteo Maran      | <i>Verificatore</i>             | Verifica del documento.<br>Causa modifiche effettuate.                                    |
| 1.04     | 2017-02-03 | Leonardo Zoso     | <i>Amministratore</i>           | Aggiunta §2.2.2.6 Progettazione di dettaglio. Causa incremento del documento.             |
| 1.03     | 2017-02-02 | Francesca Del Nin | <i>Amministratore</i>           | Aggiunta §2.2.2.5 Progettazione architetture. Causa incremento del documento.             |
| 1.02     | 2017-01-30 | Alessandro Bari   | <i>Verificatore</i>             | Verifica del documento.<br>Causa modifiche effettuate.                                    |

| Versione | Data       | Autore           | Ruolo                           | Descrizione   |
|----------|------------|------------------|---------------------------------|---|
| 1.01     | 2017-01-25 | Abdelilah Lahmer | <i>Amministratore</i>           | Correzione stile di scrittura rendendolo procedurale. Causa resoconto consegna.       |
| 1.00     | 2016-12-12 | Leonardo Zoso    | <i>Responsabile di progetto</i> | Approvazione del documento. Causa consegna del documento.                             |
| 0.18     | 2016-12-12 | Abdelilah Lahmer | <i>Verificatore</i>             | Verifica del documento. Causa modifiche effettuate.                                   |
| 0.17     | 2016-12-12 | Alessandro Bari  | <i>Amministratore</i>           | Correzione errori. Causa errori riscontrati.  |
| 0.16     | 2016-12-12 | Andrei Tabacariu | <i>Verificatore</i>             | Verifica del documento. Causa modifiche effettuate.                                   |
| 0.15     | 2016-12-9  | Alessandro Bari  | <i>Amministratore</i>           | Terminata stesura §4 Processi organizzativi. Causa incremento del documento.          |
| 0.14     | 2016-12-7  | Alessandro Bari  | <i>Amministratore</i>           | Iniziata stesura corpo dei §4 Processi organizzativi. Causa incremento del documento. |

| Versione | Data      | Autore          | Ruolo                 | Descrizione  |
|----------|-----------|-----------------|-----------------------|--|
| 0.13     | 2016-12-7 | Matteo Maran    | <i>Amministratore</i> | Aggiunta sezione §4.4 Formazione. Causa incremento del documento.  |
| 0.12     | 2016-12-7 | Matteo Maran    | <i>Amministratore</i> | Definizione struttura §4 Processi organizzativi. Causa incremento del documento.                                     |
| 0.11     | 2016-12-7 | Alessandro Bari | <i>Amministratore</i> | Terminata stesura §3 Processi di supporto. Causa incremento del documento.   |
| 0.10     | 2016-12-6 | Alessandro Bari | <i>Amministratore</i> | Integrazione §3 Processi di supporto con §3.1.1 Norme per la verifica dei documenti. Causa incremento del documento. |
| 0.09     | 2016-12-6 | Matteo Maran    | <i>Amministratore</i> | Cambiata disposizione Frontespizio e integrazione §3 Processi di supporto. Causa errori riscontrati.                 |
| 0.08     | 2016-12-5 | Alessandro Bari | <i>Amministratore</i> | Stesura iniziale §3 Processi di supporto. Causa incremento del documento.  |

| Versione | Data       | Autore            | Ruolo                 | Descrizione   |
|----------|------------|-------------------|-----------------------|---|
| 0.07     | 2016-12-5  | Francesca Del Nin | <i>Verificatore</i>   | Verifica del documento.<br>Causa modifiche effettuate.                        |
| 0.06     | 2016-12-5  | Alessandro Bari   | <i>Amministratore</i> | Terminata stesura §2<br>Processi primari.<br>Causa incremento del documento.  |
| 0.05     | 2016-12-2  | Alessandro Bari   | <i>Amministratore</i> | Stesura iniziale §2 Processi primari.<br>Causa incremento del documento.      |
| 0.04     | 2016-12-1  | Matteo Maran      | <i>Amministratore</i> | Terminata stesura §1<br>Introduzione.<br>Causa incremento del documento.      |
| 0.03     | 2016-11-30 | Matteo Maran      | <i>Amministratore</i> | Stesura iniziale §1 Introduzione.<br>Causa incremento del documento.          |
| 0.02     | 2016-11-30 | Matteo Maran      | <i>Amministratore</i> | Divisione struttura documento in sezioni.<br>Causa definizione del documento. |
| 0.01     | 2016-11-30 | Matteo Maran      | <i>Amministratore</i> | Definizione struttura generale documento.<br>Causa definizione del documento. |



## Indice

|           |  |          |
|-----------|--|----------|
| <b>1</b>  | <b>Introduzione</b>                          | <b>1</b> |
| 1.1       | Scopo del documento . . . . .                | 1        |
| 1.2       | Scopo del prodotto . . . . .                 | 1        |
| 1.3       | Glossario . . . . .                          | 2        |
| 1.4       | Riferimenti . . . . .                        | 2        |
| 1.4.1     | Riferimenti normativi . . . . .              | 2        |
| 1.4.2     | Riferimenti informativi . . . . .            | 2        |
| <b>2</b>  | <b>Processi primari</b>                      | <b>3</b> |
| 2.1       | Fornitura . . . . .                          | 3        |
| 2.1.1     | Studio di fattibilità . . . . .              | 3        |
| 2.1.2     | Collaudo . . . . .                           | 3        |
| 2.2       | Sviluppo . . . . .                           | 4        |
| 2.2.1     | Analisi dei requisiti . . . . .              | 4        |
| 2.2.1.1   | Classificazione dei casi d'uso . . . . .     | 5        |
| 2.2.1.2   | Strumento per la creazione dei diagrammi UML | 6        |
| 2.2.2     | Progettazione . . . . .                      | 6        |
| 2.2.2.1   | Diagrammi UML . . . . .                      | 6        |
| 2.2.2.2   | Requisiti per i progettisti . . . . .        | 7        |
| 2.2.2.3   | Obiettivi . . . . .                          | 7        |
| 2.2.2.4   | Tecniche di modularizzazione . . . . .       | 7        |
| 2.2.2.5   | Progettazione architetturale . . . . .       | 8        |
| 2.2.2.5.1 | Classificazione dei componenti . . . . .     | 9        |
| 2.2.2.5.2 | Test di integrazione . . . . .               | 9        |
| 2.2.2.6   | Progettazione di dettaglio . . . . .         | 9        |
| 2.2.2.6.1 | Classificazione di una classe . . . . .      | 10       |
| 2.2.2.6.2 | Descrivere una classe . . . . .              | 10       |
| 2.2.2.6.3 | Test di unità . . . . .                      | 10       |
| 2.2.2.7   | Design pattern . . . . .                     | 10       |
| 2.2.3     | Codifica . . . . .                           | 11       |
| 2.2.3.1   | Convenzioni . . . . .                        | 11       |
| 2.2.3.2   | Nomi . . . . .                               | 11       |
| 2.2.3.3   | Ricorsione . . . . .                         | 11       |
| 2.2.3.4   | Documentazione . . . . .                     | 11       |
| 2.2.3.4.1 | File . . . . .                               | 11       |
| 2.2.3.4.2 | Classi . . . . .                             | 11       |
| 2.2.3.4.3 | Metodi . . . . .                             | 11       |
| 2.2.3.4.4 | Casi eccezionali . . . . .                   | 12       |
| 2.2.3.4.5 | Versionamento dei file di codice . . . . .   | 12       |

---

|             |   |           |
|-------------|---|-----------|
| 2.2.3.4.6   | Formattazione . . . . .                   | 12        |
| <b>3</b>    | <b>Processi di supporto</b>               | <b>13</b> |
| 3.1         | Documentazione . . . . .                  | 13        |
| 3.1.1       | Norme . . . . .                           | 13        |
| 3.1.1.1     | Struttura dei documenti . . . . .         | 13        |
| 3.1.1.1.1   | Frontespizio . . . . .                    | 13        |
| 3.1.1.1.2   | Registro delle modifiche . . . . .        | 14        |
| 3.1.1.1.3   | Indice delle sezioni . . . . .            | 14        |
| 3.1.1.1.4   | Indice delle tabelle . . . . .            | 15        |
| 3.1.1.1.5   | Indice delle figure . . . . .             | 15        |
| 3.1.1.1.6   | Introduzione . . . . .                    | 15        |
| 3.1.1.1.7   | Contenuto . . . . .                       | 15        |
| 3.1.1.2     | Norme tipografiche . . . . .              | 15        |
| 3.1.1.2.1   | Formattazione generale . . . . .          | 15        |
| 3.1.1.2.1.1 | Intestazioni . . . . .                    | 15        |
| 3.1.1.2.1.2 | Piè pagina . . . . .                      | 16        |
| 3.1.1.2.2   | Caratteri . . . . .                       | 16        |
| 3.1.1.2.3   | Stile del testo . . . . .                 | 17        |
| 3.1.1.2.4   | Composizione del testo . . . . .          | 17        |
| 3.1.1.2.5   | Formati . . . . .                         | 18        |
| 3.1.1.2.6   | Nomi . . . . .                            | 19        |
| 3.1.1.2.7   | Componenti grafiche . . . . .             | 20        |
| 3.1.1.3     | Tipologie di documenti . . . . .          | 20        |
| 3.1.1.3.1   | Documenti formali . . . . .               | 20        |
| 3.1.1.3.2   | Documenti informali . . . . .             | 20        |
| 3.1.1.3.3   | Glossario . . . . .                       | 21        |
| 3.1.1.3.4   | Verbali . . . . .                         | 21        |
| 3.1.1.3.4.1 | Riunioni interne . . . . .                | 21        |
| 3.1.1.3.4.2 | Riunioni esterne . . . . .                | 21        |
| 3.1.1.3.4.3 | Struttura dei verbali . . . . .           | 22        |
| 3.1.1.3.4.4 | Tracciamento delle decisioni . . . . .    | 22        |
| 3.1.2       | Procedure . . . . .                       | 23        |
| 3.1.2.1     | Versionamento dei documenti . . . . .     | 23        |
| 3.1.2.2     | Avanzamento di un documento . . . . .     | 23        |
| 3.1.2.2.1   | Avanzamento di versione . . . . .         | 23        |
| 3.1.2.2.2   | Formalizzazione di un documento . . . . . | 23        |
| 3.1.3       | Strumenti . . . . .                       | 24        |
| 3.1.3.1     | Latex . . . . .                           | 24        |
| 3.1.3.1.1   | Comandi personalizzati . . . . .          | 24        |
| 3.2         | Qualità . . . . .                         | 24        |

---

---

|           |  |    |
|-----------|--|----|
| 3.3       | Configurazione . . . . .                               | 24 |
| 3.3.1     | Norme . . . . .  | 24 |
| 3.3.1.1   | Controllo di versione . . . . .                        | 24 |
| 3.3.1.2   | Richieste di modifica . . . . .                        | 25 |
| 3.3.1.3   | Repository . . . . .                                   | 25 |
| 3.3.1.3.1 | Struttura del repository . . . . .                     | 25 |
| 3.3.1.3.2 | Commit . . . . .                                       | 26 |
| 3.3.1.3.3 | Visibilità del repository . . . . .                    | 26 |
| 3.3.1.4   | File . . . . .   | 26 |
| 3.3.1.4.1 | Nomi dei file . . . . .                                | 26 |
| 3.3.2     | Procedure . . . . .                                    | 27 |
| 3.3.2.1   | Richiesta di modifica . . . . .                        | 27 |
| 3.3.2.2   | Aggiornamento del repository . . . . .                 | 28 |
| 3.3.3     | Strumenti . . . . .                                    | 28 |
| 3.4       | Verifica . . . . .                                     | 29 |
| 3.4.1     | Documenti . . . . .                                    | 29 |
| 3.4.1.1   | Sintassi . . . . .                                     | 29 |
| 3.4.1.2   | Struttura del documento . . . . .                      | 29 |
| 3.4.1.3   | Resoconto attività di verifica sui documenti . . . . . | 29 |
| 3.4.1.4   | Attività manuale di verifica . . . . .                 | 30 |
| 3.4.1.5   | Attività automatica di verifica . . . . .              | 30 |
| 3.4.2     | Diagrammi UML . . . . .                                | 30 |
| 3.4.2.1   | Diagrammi dei casi d'uso . . . . .                     | 31 |
| 3.4.3     | Issue tracking . . . . .                               | 31 |
| 3.4.3.1   | Sintassi di una label . . . . .                        | 31 |
| 3.4.3.2   | Sintassi di una Issue . . . . .                        | 31 |
| 3.4.3.3   | Gestione di una issue . . . . .                        | 32 |
| 3.4.3.4   | Strumento per l'issue tracking . . . . .               | 33 |
| 3.4.4     | Tecniche di analisi . . . . .                          | 33 |
| 3.4.4.1   | Analisi statica . . . . .                              | 33 |
| 3.4.4.2   | Analisi dinamica . . . . .                             | 34 |
| 3.4.5     | Test . . . . .   | 34 |
| 3.4.5.1   | Classificazione dei test . . . . .                     | 34 |
| 3.5       | Validazione . . . . .                                  | 35 |
| 3.5.1     | Responsabilità . . . . .                               | 35 |
| 3.5.2     | Procedure . . . . .                                    | 36 |
| 3.5.2.1   | Procedura per la validazione . . . . .                 | 36 |

---

|           |  |           |
|-----------|--|-----------|
| <b>4</b>  | <b>Processi organizzativi</b>                          | <b>37</b> |
| 4.1       | Norme . . . . .  | 37        |
| 4.1.1     | Comunicazioni . . . . .                                | 37        |
| 4.1.1.1   | Interne . . . . .                                      | 37        |
| 4.1.1.2   | Esterne . . . . .                                      | 37        |
| 4.1.1.3   | Composizione delle email . . . . .                     | 37        |
| 4.1.1.3.1 | Mittente . . . . .                                     | 38        |
| 4.1.1.3.2 | Destinatario . . . . .                                 | 38        |
| 4.1.1.3.3 | Oggetto . . . . .                                      | 38        |
| 4.1.1.3.4 | Corpo . . . . .  | 38        |
| 4.1.1.3.5 | Allegati . . . . .                                     | 39        |
| 4.1.2     | Riunioni . . . . .                                     | 39        |
| 4.1.2.1   | Interne . . . . .                                      | 39        |
| 4.1.2.1.1 | Convocazione . . . . .                                 | 39        |
| 4.1.2.1.2 | Gestione . . . . .                                     | 40        |
| 4.1.2.2   | Esterne . . . . .                                      | 40        |
| 4.1.2.2.1 | Convocazione . . . . .                                 | 40        |
| 4.1.2.2.2 | Gestione . . . . .                                     | 40        |
| 4.1.2.3   | Verbale . . . . .                                      | 41        |
| 4.1.3     | Ruoli di progetto . . . . .                            | 41        |
| 4.1.3.1   | Responsabile di Progetto . . . . .                     | 41        |
| 4.1.3.2   | Amministratore . . . . .                               | 42        |
| 4.1.3.3   | Analista . . . . .                                     | 42        |
| 4.1.3.4   | Progettista . . . . .                                  | 43        |
| 4.1.3.5   | Programmatore . . . . .                                | 43        |
| 4.1.3.6   | Verificatore . . . . .                                 | 44        |
| 4.1.3.7   | Rotazione dei ruoli . . . . .                          | 44        |
| 4.1.4     | Ticketing . . . . .                                    | 44        |
| 4.2       | Procedure . . . . .                                    | 45        |
| 4.2.1     | Creazione e gestione di un ticket . . . . .            | 45        |
| 4.3       | Strumenti . . . . .                                    | 45        |
| 4.3.1     | Pianificazione . . . . .                               | 45        |
| 4.3.2     | Tracciamento dei requisiti, test, casi d'uso . . . . . | 46        |
| 4.3.3     | Calcolo del consuntivo . . . . .                       | 46        |
| 4.3.4     | Meccanismi di controllo . . . . .                      | 47        |
| 4.3.4.0.1 | Controllo metriche di progetto . . . . .               | 47        |
| 4.3.4.0.2 | Meccanismi di rendicontazione . . . . .                | 47        |
| 4.3.5     | Lista di controllo . . . . .                           | 48        |
| 4.4       | Formazione dei membri del gruppo . . . . .             | 48        |

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di:

1. stabilire le norme che i membri del gruppo<sub>G</sub> *Answer* devono rispettare durante lo sviluppo del progetto AtAVi: Accoglienza tramite Assistente Virtuale<sub>G</sub>;
2. gli strumenti che devono utilizzare;
3. le procedure che devono seguire.

Tutti i membri del gruppo sono tenuti a prendere visione di tale documento e rispettare le norme in esso contenute, al fine di garantire uniformità nello svolgimento del progetto.

Il documento rivolge l'attenzione sui seguenti contenuti:

- definizione dei ruoli per ogni membro del gruppo;
- organizzazione della comunicazione interna ed esterna;
- definizione dell'ambiente di sviluppo;
- modalità di gestione e utilizzo del repository;
- modalità di lavoro e di stesura dei documenti.

## 1.2 Scopo del prodotto

Il prodotto richiesto è un assistente virtuale<sub>G</sub> che svolge la funzione di accoglienza clienti in un'azienda. Il prodotto comprenderà:

- un'interfaccia Web<sub>G</sub> di interazione con l'utente utilizzatore del sistema;
- gestione dei servizi *AWS Lambda*<sub>G</sub> per l'interazione con le API<sub>G</sub> dell'assistente virtuale;
- un interfacciamento con *Slack*<sub>G</sub>, in modo da poter notificare gli interessati sull'arrivo del cliente.

## 1.3 Glossario

Il glossario costituirà un documento esterno che dovrà fornire le definizioni per alcuni termini che compaiono nei documenti, in particolare:

- termini tecnici;
- termini ambigui;
- abbreviazioni.

La prima occorrenza di un termine che è presente in *Glossario-v.4.00* deve essere seguita dalla marcatura pedice <sub>g</sub>.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- capitolati<sub>g</sub> d'appalto: <http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/Capitolati.html>.

### 1.4.2 Riferimenti informativi

- Materiale del corso di Ingegneria del Software:
  - Premesse:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L01.pdf>;
  - Gestione di Progetto:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L04.pdf>;
  - Amministrazione di Progetto:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L05.pdf>;
  - Progettazione Software:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L07.pdf>.
- Materiale esterno al corso:
  - Siri kit:** <http://developer.apple.com/sirikit/>;
  - Alexa skills kit:** <http://developer.amazon.com/alexa-skills-kit>;
  - Avvertenze:** §*Avvertenze* nel documento *DefinizioneDiProdotto-v.3.00*.

## 2 Processi primari

§

### 2.1 Fornitura

#### 2.1.1 Studio di fattibilità

Il documento riguardante lo *Studio di fattibilità* deve essere redatto in modo accurato e tempestivo dagli analisti dopo la visione dei capitolati e dopo una serie di riunioni in merito. Il documento dovrà fornire informazioni quantitative e dovrà trattare i seguenti argomenti per ogni capitolato:

- **Rischi** nell'affrontare i progetti richiesti;
- **Descrizione** generale di ciò che viene richiesto nei capitolati;
- **Rapporto** tra i costi ed i benefici;
- **Dominio** applicativo e tecnologico richiesto per ogni progetto;
- **Valutazione finale** delle criticità.

#### 2.1.2 Collaudo

In preparazione al collaudo devono essere effettuati con successo i test presentati in §3.4.5, in particolare:

- test di unità;
- test di integrazione;
- test di sistema;
- test di accettazione.

Devono essere effettuate le misurazioni delle metriche riportate in §2, §3 del documento *PianoDiQualifica\_v.4.00*, ottenendo dei valori che rientrino nel *range accettato*. Devono essere attuate le politiche di verifica (§3.4) e validazione (§3.5) sui documenti:

- *ManualeUtente\_v.2.00*;
- *PianoDiProgetto\_v.4.00*;
- *PianoDiQualifica\_v.4.00*;
- sui documenti che hanno subito incrementi nel periodo di Validazione.

## 2.2 Sviluppo

### 2.2.1 Analisi dei requisiti

L'*Analisi dei requisiti* è un documento che raccoglie e definisce tutti i requisiti che il prodotto finale dovrà soddisfare. L'operazione di analisi per individuare i requisiti è composta da più fasi:

**Studio dei bisogni e delle fonti:** utile per identificare, specificare e classificare i requisiti;

**Classificazione dei requisiti:** i requisiti vengono classificati in base al tipo e alla priorità, utilizzando la seguente notazione:

$$R[X][Y][Z]$$

dove:

1. **X** indica la tipologia del requisito. Deve assumere solo i seguenti valori:
  - **F**: indica un requisito funzionale;
  - **Q**: indica un requisito di qualità;
  - **P**: indica un requisito prestazionale;
  - **V**: indica un requisito vincolo.
2. **Y** indica l'importanza strategica del requisito. Deve assumere solo i seguenti valori:
  - **Obb**: indica un requisito obbligatorio;
  - **Des**: indica un requisito desiderabile;
  - **Opz**: indica un requisito opzionale.
3. **Z** rappresenta il codice univoco di ogni requisito in forma gerarchica.

**Modellazione concettuale del sistema:** partizionamento in componenti per consentire l'allocazione dei requisiti (vengono utilizzati i diagrammi dei casi d'uso);

**Assegnazione dei requisiti a parti distinte del sistema:** ogni requisito, dopo essere stato studiato e classificato, viene assegnato ad un membro del gruppo in base alle sue competenze;

**Negoziazione con il committente e con i sotto-fornitori:** i requisiti, specie i più complessi, possono essere negoziati e modificati con il committente.



Per ogni requisito si dovrà inoltre indicare:

- **Descrizione:** una breve descrizione meno ambigua possibile;
- **Fonte:** la fonte può essere una delle seguenti:
  - Capitolato: deriva direttamente dal testo del capitolato;
  - Verbale: deriva da un incontro verbalizzato;
  - Interno: deriva da discussioni interne al gruppo;
  - Casi d'uso: deriva da uno o più casi d'uso.

**2.2.1.1 Classificazione dei casi d'uso** È compito degli analisti redigere una descrizione, dare una classificazione e fornire un diagramma conforme allo standard UML<sub>g</sub> per ogni caso d'uso. Ogni caso d'uso dev'essere descritto con le seguenti informazioni:

1. Codice identificativo del caso d'uso, nella forma

**UC[Codice padre].[Codice identificativo]**

dove:

- **Codice padre** è il codice univoco del padre;
  - **Codice identificativo** è un codice progressivo di livello.
2. **Titolo**, che deve descrivere sinteticamente il caso d'uso;
  3. **Attori principali**;
  4. **Attori secondari**, se sono presenti;
  5. **Precondizioni**, ovvero le condizioni che necessariamente devono verificarsi prima del caso d'uso;
  6. **Postcondizioni**, ovvero le condizioni che devono verificarsi successivamente al caso d'uso;
  7. **Flusso principale** degli eventi, dove si descrive il flusso dei casi d'uso figli.

Per ogni evento va specificato:

- una descrizione testuale dell'evento;
- gli attori coinvolti;

- se l'azione è descritta dettagliatamente da un altro caso d'uso.

8. **Flussi alternativi**, ovvero scenari in cui si verificano eccezioni o errori. Per ognuno di questi deve essere indicato:

- una descrizione testuale dell'evento;
- gli attori coinvolti;
- se l'azione è descritta dettagliatamente da un altro caso d'uso.

**2.2.1.2 Strumento per la creazione dei diagrammi UML** Lo strumento per la creazione dei diagrammi UML che il gruppo utilizzerà per compiere l'analisi dei requisiti è *Astah<sub>G</sub>*.

## 2.2.2 Progettazione

Lo scopo dell'attività di progettazione è generare una soluzione soddisfacente per tutti i membri del gruppo. Una volta compresi pienamente quali siano i requisiti richiesti dal problema e una volta approfondita la progettazione in moduli abbastanza semplici da essere capiti da una sola persona, si otterranno le istruzioni necessarie ai *Programmatori* per sviluppare il prodotto finito.

**2.2.2.1 Diagrammi UML** Per documentare prodotti e componenti individuati, sarà necessario ricorrere a dei diagrammi UML, necessari per formalizzare gli aspetti descritti in modo testuale. Durante la progettazione architettuale, sarà necessario ricorrere a diagrammi dei componenti che hanno i seguenti scopi:

- rappresentare la struttura interna di un prodotto software modellato in termini dei suoi componenti principali e delle relazioni fra di essi;
- evidenziare i componenti (e le relazioni che essi hanno fra di loro) individuati all'interno di ciascun prodotto software.

La progettazione deve utilizzare le seguenti tipologie di diagrammi UML:

- **Diagrammi dei package:** raggruppamenti di classi in una unità di livello più alto;
- **Diagrammi di attività:** hanno i seguenti scopi:
  - definire le attività da svolgere per realizzare una certa funzionalità;

- mostrare come determinate interazioni tra componenti (o tra prodotti) realizzino una funzionalità che si intende rendere disponibile.

- **Diagrammi di sequenza:** hanno i seguenti scopi:

- descrivere scenari;
- descrivere le relazioni che intercorrono, in termini di messaggi, tra attori, oggetti ed entità del sistema rappresentato.

**2.2.2.2 Requisiti per i progettisti** I *Progettisti* sono responsabili delle attività di progettazione e, per garantire la qualità del prodotto, sono tenuti ad avere:

- profonda conoscenza di tutto ciò che riguarda il processo di sviluppo del software;
- capacità di saper anticipare i cambiamenti;
- notevole inventiva per riuscire a trovare una soluzione progettuale accettabile anche in mancanza di una metodologia che sia sufficientemente espressiva;
- capacità di individuare con rapidità e sicurezza le soluzioni più opportune.

**2.2.2.3 Obiettivi** La fase di progettazione si pone i seguenti obiettivi:

- progettare un software con le caratteristiche di qualità specificate nella fase di analisi e specifica dei requisiti;
- capacità di poter far fronte a modifiche senza che l'intera struttura del software già costruita venga messa in discussione;
- soddisfare i requisiti di qualità fissati.

**2.2.2.4 Tecniche di modularizzazione** I *Progettisti* devono definire delle *tecniche di modularizzazione* per garantire la qualità del prodotto richiesta. Le *tecniche di modularizzazione* consentono:

- la scomposizione del sistema in moduli;
- la descrizione precisa della struttura modulare e delle relazioni che esistono tra i singoli moduli.

L'attività di modularizzazione consentirà i seguenti vantaggi:

- semplificazione dell'attività di verifica;
- riusabilità del software;
- leggibilità del codice;
- possibilità di realizzazione di prototipi;
- semplificazione dell'attività di manutenzione.

**2.2.2.5 Progettazione architetturale** Per poter arrivare alla definizione dell'architettura dei vari prodotti software, è necessario passare attraverso una serie di attività, in particolare:

1. individuare i prodotti che si intendono realizzare;
2. definire ruoli e responsabilità per ogni prodotto individuato;
3. definire le interazioni che i prodotti hanno fra di loro;
4. assicurarsi che ogni requisito sia soddisfatto da almeno uno dei prodotti individuati;
5. ci si assicura che ogni prodotto soddisfi almeno un requisito.

A questo punto, è possibile procedere con la progettazione dell'architettura dei prodotti software che devono essere realizzati. Si devono eseguire e documentare i seguenti task per ogni prodotto individuato:

1. suddividere il prodotto in componenti;
2. definire il ruolo di ogni componente individuato;
3. definire le interazioni tra i vari componenti;
4. definire le interfacce che ogni componente mette a disposizione;
5. assicurarsi che ogni requisito sia soddisfatto da almeno uno dei componenti;
6. assicurarsi che ogni componente soddisfi almeno un requisito;
7. realizzare e documentare i test di integrazione, al fine di verificare il corretto funzionamento di più componenti integrati assieme.

Durante la suddivisione del prodotto in più componenti i *Progettisti* dovranno utilizzare i design pattern e, nel farlo, dovranno provvedere a descriverli.

**2.2.2.5.1 Classificazione dei componenti** I componenti vengono identificati univocamente da una descrizione nella forma

**NomeProdotto::NomeComponente**

dove:

- NomeProdotto è il nome del prodotto software che i *Progettisti* hanno individuato;
- NomeComponente rappresenta il nome assegnato al componente dai *Progettisti*.

Inoltre, ad ogni componente è associato un codice univoco nella forma [X][Y], dove:

- X è l'iniziale del nome del prodotto;
- Y è un numero intero incrementale.

**2.2.2.5.2 Test di integrazione** Per ottimizzare l'attività di test, i *Progettisti* devono preoccuparsi di definire delle classi di verifica per accertare il funzionamento dei componenti fornendo, se possibile, degli strumenti automatici ai Verificatori. La progettazione delle classi per la verifica deve essere svolta nel rispetto della mancata sovrapposizione dei ruoli: chi crea una classe non dovrebbe essere lo stesso individuo che deve testarla.

**2.2.2.6 Progettazione di dettaglio** Una volta definita l'architettura del sistema, si procede alla progettazione di dettaglio, la quale prevede i seguenti task:

1. individuare le classi che implementano ciascuno dei componenti individuati mediante la progettazione architetturale;
2. definire i ruoli e le responsabilità di ogni classe individuata;
3. definire nel dettaglio ogni classe;
4. assicurarsi che ogni requisito sia soddisfatto da almeno una delle classi definite;
5. assicurarsi che ogni classe soddisfi almeno un requisito;
6. realizzare e documentare i test di unità, necessari per verificare il corretto comportamento di ogni classe.

La definizione di classe viene spiegata meglio in §1.5 del documento *DefinizioneDiProdotto\_v.3.00*.

**2.2.2.6.1 Classificazione di una classe** Le classi vengono univocamente identificate da una descrizione nella forma

**NomeProdotto::NomeComponente::NomeClasse**

dove:

- NomeProdotto è il nome del prodotto software che i *Progettisti* hanno individuato;
- NomeComponente rappresenta il nome assegnato al componente dai *Progettisti*;
- NomeClasse rappresenta il nome che è stato dato dai *Progettisti* alla classe individuata.

**2.2.2.6.2 Descrivere una classe** Per descrivere una classe, i *Progettisti* devono preoccuparsi di descrivere:

- nome;
- visibilità;
- attributi;
- metodi;
- descrizione generica che indichi lo scopo e la responsabilità della classe.

Inoltre, devono essere descritte dettagliatamente (anche attraverso diagrammi) le relazioni con altre classi e le interfacce messe a disposizione.

**2.2.2.6.3 Test di unità** Per la realizzazione dei test di unità, i *Progettisti* devono rispettare la regola di mancata sovrapposizione dei ruoli, in particolare chi realizza il test non deve essere lo stesso individuo che ha realizzato la classe da testare.

**2.2.2.7 Design pattern** Per migliorare la comprensibilità delle scelte progettuali e della progettazione stessa i *Progettisti* dovranno indicare i pattern architetturali utilizzati, fornendo per ciascuno d'essi:

- descrizione testuale;
- motivazione dell'utilizzo;
- descrizione di come viene applicato il pattern al progetto.

### 2.2.3 Codifica

**2.2.3.1 Convenzioni** Tutti i file contenenti codice o testo dovranno rispettare la codifica UTF-8 senza BOM.

**2.2.3.2 Nomi** I nomi di variabili, metodi e classi dovranno essere in notazione CamelCase e in lingua inglese. I nomi di variabili e metodi dovranno avere la prima lettera minuscola, mentre per le classi sarà maiuscola.

**2.2.3.3 Ricorsione** La ricorsione dovrà essere evitata il più possibile, limitando il suo uso ai soli casi indispensabili, ossia dove sia dimostrato che non è possibile utilizzare un metodo iterativo con complessità computazionale minore. Per ogni metodo ricorsivo, dovrà essere fornita una prova della sua terminazione.

#### 2.2.3.4 Documentazione

**2.2.3.4.1 File** I file contenenti codice dovranno avere un'intestazione con le seguenti componenti:

- nome dell'autore;
- versione corrente del file;
- versione del file nel momento dell'aggiunta al progetto;
- descrizione del file.

**2.2.3.4.2 Classi** Ogni classe deve presentare la sua descrizione nell'intestazione.

**2.2.3.4.3 Metodi** Per ogni metodo, dovrà essere presente un'intestazione contenente:

- descrizione del metodo;
- descrizione del primo parametro;
- descrizione del N-esimo parametro;
- valore ritornato dal metodo;
- motivo di lancio dell'eccezione.

**2.2.3.4.4 Casi eccezionali** Qualora fosse necessario documentare parti di codice di difficile comprensione è permesso l'utilizzo di un commento nelle righe precedenti che presenti la descrizione del blocco di codice sottostante.

**2.2.3.4.5 Versionamento dei file di codice** Il numero di versione dei file di codice appare solamente nell'intestazione di ciascuno di essi. Deve seguire il formato:

$$[X].[Y]$$

dove:

- X è un numero intero e incrementale corrispondente all'ultima versione stabile del codice;
- Y è un numero intero e incrementale corrispondente al numero di modifiche apportate al codice, partendo dall'ultima versione stabile.

**2.2.3.4.6 Formattazione** La formattazione facilita la comprensione dell'organizzazione logica del codice. Per consentire agli sviluppatori di decifrare il codice sorgente è fondamentale una formattazione logica e coerente. Di seguito sono elencati alcuni metodi di formattazione:

- definire una dimensione standard per i rientri e utilizzarla in maniera coerente;
- allineare le sezioni di codice utilizzando il rientro predefinito;
- rientrare le righe di codice secondo la relativa costruzione logica, se non viene utilizzato il rientro, il codice risulterà di difficile comprensione;
- utilizzare spazi prima e dopo la maggior parte degli operatori, se ciò non altera la funzione del codice;
- suddividere logicamente il codice sorgente tra diversi file fisici;
- suddividere le sezioni complesse ed estese di codice in moduli comprensibili di dimensioni minori.



## 3 Processi di supporto

### 3.1 Documentazione

In questa sezione sono indicati gli standard riguardanti la struttura e la stesura dei documenti prodotti.

#### 3.1.1 Norme

**3.1.1.1 Struttura dei documenti** Ad eccezione dei verbali, ogni documento dovrà utilizzare una struttura prestabilita ed ordinata in questo modo:

1. frontespizio;
2. informazioni sul documento;
3. registro delle modifiche;
4. indice delle sezioni;
5. (eventuale) indice delle tabelle;
6. (eventuale) indice delle figure;
7. introduzione;
8. contenuto.

**3.1.1.1.1 Frontespizio** Questa sezione deve contenere:

1. intestazione con il nome del prodotto;
2. nome del documento;
3. logo del gruppo;
4. logo del proponente;
5. email del gruppo;
6. informazioni sul documento:
  - nome;
  - versione;
  - data di creazione;

- data ultima modifica;
- stato (approvato/non approvato);
- redazione, cioè chi lo ha scritto;
- verifica, cioè chi lo ha verificato;
- approvazione, cioè chi lo ha approvato;
- distribuzione, indica il nome del gruppo;
- destinato a;
- email del gruppo.

7. sommario, indica l'oggetto del documento.

**3.1.1.1.2 Registro delle modifiche** Questa sezione tabellare riporta le modifiche effettuate al documento. Ogni riga della tabella corrisponde ad una versione diversa del documento, poste in ordine cronologico discendente, e la struttura di ogni riga deve essere la seguente:

1. versione del documento;
2. data della modifica;
3. autore della modifica;
4. ruolo dell'autore della modifica nel momento in cui essa è avvenuta;
5. breve descrizione delle modifiche apportate.

**3.1.1.1.3 Indice delle sezioni** L'indice delle sezioni deve contenere l'indice e la pagina di tutti gli argomenti trattati all'interno del documento. La struttura utilizzata per l'indice viene generata automaticamente da  $\text{\LaTeX}$  ed è la seguente:

1. indice numerico;
2. titolo dell'argomento trattato;
3. numero della pagina.

**3.1.1.1.4 Indice delle tabelle** Questa sezione contiene l'indice delle tabelle e non compare in caso non ci siano tabelle nel documento. Per ogni tabella deve essere specificato:

1. indice numerico;
2. titolo della tabella;
3. numero della pagina.

**3.1.1.1.5 Indice delle figure** In questa sezione sono riportate tutte le eventuali figure presenti all'interno del documento. Per ogni figura deve essere specificato:

1. indice numerico;
2. nome figura;
3. numero della pagina.

**3.1.1.1.6 Introduzione** Questa sezione deve riportare un riassunto generale del documento con alcuni temi obbligatori:

1. scopo del documento;
2. riferimenti normativi;
3. riferimenti informativi.

**3.1.1.1.7 Contenuto** Questa sezione è la più importante e complessa di tutto il documento ed essendo molto lunga, deve essere divisa in sezioni, sottosezioni e paragrafi.

### **3.1.1.2 Norme tipografiche**

#### **3.1.1.2.1 Formattazione generale**

**3.1.1.2.1.1 Intestazioni** Ogni pagina di un documento deve contenere l'intestazione, composta da:

- logo del gruppo, posizionato a sinistra;
- nome del documento posizionato a destra.

**3.1.1.2.1.2 Piè pagina** Deve contenere i seguenti indici:

- numero romano della pagina corrente e numero romano dell'ultima pagina per la parte di frontespizio, indici vari e registro delle modifiche;
- numero arabo della pagina corrente e numero arabo dell'ultima pagina, per la parte di contenuto.

**3.1.1.2.2 Caratteri**

- ‘§’, indica il riferimento ad una sezione di un documento;
- Virgolette:
  - **Virgolette alte singole** ‘ ’ : devono essere utilizzate per racchiudere un singolo carattere;
  - **Virgolette alte doppie** “ ” : devono essere utilizzate per racchiudere:
    - \* nomi di file;
    - \* comandi;
    - \* parole a cui è stato dato un significato particolare;
    - \* parole a cui è stato dato un senso diverso da quello originale.
- Parentesi
  - **Tonde:** possono essere utilizzate per descrivere esempi, per fornire dei sinonimi oppure per dare delle precisazioni. Sono le uniche parentesi ammesse all'interno di una frase;
  - **Quadre:** rappresentano uno standard ISO<sub>c</sub>.
- Numeri
  - la parte frazionaria di un numero deve essere separata da quella decimale utilizzando la virgola;
  - i numeri la cui parte intera supera le tre cifre, devono essere scritti raggruppando in gruppi di tre le cifre, partendo dalla cifra meno significativa e separandoli con uno spazio.
- Costi
  - la parte frazionaria di un costo deve essere separata da quella decimale utilizzando la virgola;

- i costi la cui parte intera supera le tre cifre, devono essere scritti raggruppando in gruppi di tre le cifre, partendo dalla cifra meno significativa e separandoli con un punto.

### 3.1.1.2.3 Stile del testo

- **Corsivo** va utilizzato per distinguere le seguenti informazioni:
  1. nome di un documento;
  2. nome di un ruolo;
  3. percorsi di cartelle;
  4. nomi di strumenti e software.
- **Grassetto** va utilizzato per distinguere le seguenti informazioni:
  - titoli;
  - parole su cui è utile focalizzare l'attenzione del lettore all'interno di un argomento;
  - parole chiave all'interno di elenchi.
- **Sottolineato** è indicato qualora si voglia evidenziare l'importanza di una parola all'interno di una frase.
- **Glossario** è una formattazione speciale, tramite un pedice contenente il carattere 'G', che segue i termini che sono spiegati nel *Glossario\_v.4.00*. Non si applica questa regola nei casi in cui la parola compaia all'interno di titoli, percorsi, nomi di cartelle, comandi o parti di codice.

### 3.1.1.2.4 Composizione del testo

- **Elenchi** Le norme che regolano un elenco sono le seguenti:
  - ogni elemento dell'elenco deve terminare con il carattere ';;
  - l'ultimo elemento di un elenco deve sempre terminare con il carattere '.'

È necessario usare elenchi **numerati** quando l'ordine degli elementi è rilevante o quando si parla di un numero fisso di componenti di un insieme.

Gli elenchi **puntati** servono per descrivere elementi di cui non è importante l'ordine espositivo.

### 3.1.1.2.5 Formati

- **Date:** La formattazione delle date segue lo standard [ISO 8601<sub>G</sub>]. Tale standard prevede che una data sia scritta secondo il seguente formalismo: YYYY-MM-DD. Questa rappresentazione va letta nel seguente modo:
  - YYYY: numero a quattro cifre che rappresenta l'anno;
  - MM: numero a due cifre che rappresenta il mese;
  - DD: numero a due cifre che rappresenta il giorno.
- **Orari:** La formattazione degli orari segue lo standard [ISO 8601<sub>G</sub>]. Tale standard prevede che gli orari siano scritti secondo il seguente formalismo: hh:mm. Questa rappresentazione va letta nel seguente modo:
  - hh: numero a due cifre che rappresenta il numero di ore trascorse dalla mezzanotte;
  - mm: numero a due cifre che rappresenta i minuti.
- **Sigle:** È possibile fare riferimento a ruoli, documenti e revisioni pianificate utilizzando le seguenti sigle:
  - Rp (responsabile di progetto);
  - Am (amministratore);
  - An (analista);
  - Pt (progettista);
  - Pm (programmatore);
  - Ve (verificatore);
  - AR (analisi dei requisiti);
  - GL (glossario);
  - NP (norme di progetto);
  - PP (piano di progetto);
  - PQ (piano di qualifica);
  - SF (studio di fattibilità);
  - ST (specificazione tecnica);
  - DP (definizione di prodotto);

- MU (manuale utente);
  - RR (revisione dei requisiti);
  - RA (revisione di accettazione);
  - RP (revisione di progettazione);
  - RQ (revisione di qualifica).
- **Ruoli di progetto:** quando si fa riferimento ad un ruolo di progetto bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola;
  - **Fasi del progetto:** quando si fa riferimento ad una fase<sub>g</sub> del progetto bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola;
  - **Revisioni:** quando si fa riferimento ad una revisione bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola.

#### 3.1.1.2.6 Nomi

- **Nome del gruppo:** quando si fa riferimento al nome del gruppo bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola;
- **Nome del progetto:** per il nome del progetto deve essere sempre usato lo stile corsivo e la prima lettera deve essere maiuscola;
- **Nome proprio:** deve comparire prima il nome e successivamente il cognome. Quando si fa riferimento ad una disposizione in ordine alfabetico, i nomi devono essere scritti mettendo il cognome davanti al nome e l'ordine è dettato dalle lettere del cognome, a meno di ulteriori specificazioni;
- **Nome di un file:** i nomi dei file vanno formattati utilizzando lo stile corsivo e devono essere racchiusi dalle doppie virgolette alte;
- **Nome di un documento:** quando si fa riferimento ad un documento bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola. I nomi dei documenti devono essere racchiusi dalle doppie virgolette alte.

### 3.1.1.2.7 Componenti grafiche

- **Immagini:** L'utilizzo delle immagini all'interno di un documento è regolamentato secondo quanto segue:
  - i formati ammessi per le immagini sono il PNG<sub>g</sub> e il PDF<sub>g</sub>;
  - devono essere numerate in ordine crescente;
  - devono essere seguite da una breve descrizione;
  - deve essere presente un riferimento all'immagine all'interno dell'indice immagini.
- **Tabelle:** L'utilizzo delle tabelle all'interno di un documento è regolamentato secondo quanto segue:
  - devono essere numerate in ordine crescente;
  - devono essere seguite da una breve descrizione;
  - deve essere presente un riferimento alla tabella all'interno dell'indice delle tabelle;
  - si consigliano le linee verticali all'interno delle tabelle solo se queste ne dovessero aumentare la leggibilità.

### 3.1.1.3 Tipologie di documenti

**3.1.1.3.1 Documenti formali** I documenti formali possono essere descritti secondo quanto segue:

- sono documenti approvati dal *Responsabile di progetto*;
- eventuali modifiche ad un documento formale, lo rendono informale;
- sono gli unici documenti che possono essere distribuiti all'esterno del gruppo di progetto.

**3.1.1.3.2 Documenti informali** I documenti informali possono essere descritti secondo quanto segue:

- sono documenti non ancora approvati dal *Responsabile di progetto*;
- possono essere distribuiti solamente all'interno del gruppo di progetto;
- possono essere sottoposti a revisione.



**3.1.1.3.3 Glossario** Il glossario nasce dall'esigenza di chiarire il significato ambiguo che possono avere certe parole all'interno di determinati contesti. Al suo interno saranno quindi presenti alcune parole, prese dai documenti, che hanno le seguenti caratteristiche:

- trattano argomenti tecnici;
- trattano argomenti poco conosciuti o che possono scatenare ambiguità;
- rappresentano delle sigle.

Il glossario deve essere strutturato secondo quanto segue:

- i termini devono seguire l'ordine lessicografico;
- ogni termine deve essere seguito da una spiegazione chiara e concisa del significato del termine stesso.

Per evitare confusione, la stesura del glossario deve avvenire in maniera parallela alla stesura dei documenti. Al fine di evitare dimenticanze, è ammesso inserire un termine all'interno del glossario senza inserirne immediatamente la spiegazione. È comunque doveroso completare la spiegazione non appena possibile.

**3.1.1.3.4 Verbali** Lo scopo di un verbale è di riassumere, cercando di essere il più possibile fedeli, ciò che è stato discusso durante un incontro. Le norme che un verbale deve seguire sono le seguenti:

- deve essere redatto ad ogni riunione tra i membri del gruppo e/o soggetti esterni;
- è soggetto ad un'unica stesura e non può subire modifiche.

**3.1.1.3.4.1 Riunioni interne** Si definisce interna una riunione che coinvolge solamente i membri del gruppo. Il verbale per questo tipo di incontro è da considerarsi di carattere informale.

**3.1.1.3.4.2 Riunioni esterne** Si definisce esterna una riunione che avviene tra i membri del gruppo e soggetti esterni. Il verbale redatto per questo tipo di incontro è da considerarsi come parte integrante della documentazione ufficiale e per questo motivo può avere un valore normativo o fornire nuovi requisiti. È previsto che ad ogni incontro venga nominata una persona che si occupi della sua stesura.

**3.1.1.3.4.3 Struttura dei verbali** I verbali, interni ed esterni devono essere redatti seguendo la seguente struttura:

1. frontespizio;
2. una sezione “Informazioni sul Verbale” contenente le seguenti informazioni:
  - (a) nome documento;
  - (b) data di creazione;
  - (c) data ultima modifica;
  - (d) redazione;
  - (e) uso;
  - (f) distribuzione;
  - (g) destinato a;
  - (h) email di riferimento.
3. sommario;
4. indice;
5. informazioni generali;
6. riassunto della riunione.

**3.1.1.3.4.4 Tracciamento delle decisioni** Per garantire l’attuazione delle decisioni, prese e descritte nei vari verbali, è necessario accompagnare ogni decisione da un codice identificativo del tipo:

**DX.YY**

dove:

- **D** può assumere i valori:
  - **E** se si riferisce ad un verbale esterno;
  - **I** se si riferisce ad un verbale interno.
- **X** indica il numero identificativo del verbale associato;
- **YY** è un numero a due cifre incrementale che indica il numero della decisione all’interno del verbale.

### 3.1.2 Procedure

**3.1.2.1 Versionamento dei documenti** Tutti i documenti, ad eccezione dei verbali, sono sottoposti a versionamento. Il versionamento prevede che la versione di un documento venga incrementata ad ogni modifica al documento stesso. La sintassi che indica la versione di un documento che ha questa forma vX.YY viene interpretata in questo modo:

- 'v' è un carattere che si riferisce alla parola versione;
- X è un numero che indica quante volte è stato formalizzato il documento;
- YY è un numero a due cifre che indica quante modifiche sono state effettuate al documento dalla sua ultima formalizzazione.

#### 3.1.2.2 Avanzamento di un documento

**3.1.2.2.1 Avanzamento di versione** L'avanzamento di versione avviene secondo le seguenti regole:

- X inizia da 0 e viene incrementato di una unità nel momento in cui il *Responsabile di progetto* formalizza il documento;
- YY inizia da 00 e viene incrementato di una unità ad ogni modifica che viene effettuata al documento. Ogni volta che il documento viene formalizzato riparte da 00.

La prima versione di ogni documento è indicata dalla versione v0.01.

**3.1.2.2.2 Formalizzazione di un documento** La formalizzazione di un documento segue la seguente procedura:

1. il documento viene redatto da coloro che sono incaricati della stesura ed eventuale correzione;
2. il *Responsabile di progetto* assegna uno o più verificatori al documento, i quali dovranno occuparsi di controllare la correttezza del documento stesso;
3. se i verificatori riscontrano anomalie si ritorna al punto 1 altrimenti il documento viene consegnato al *Responsabile di progetto*;
4. il *Responsabile di progetto* decide se approvare il documento, e quindi formalizzarlo, oppure se rifiutarlo e quindi ritornare al punto 1.

### 3.1.3 Strumenti

**3.1.3.1 Latex** La stesura dei documenti deve essere effettuata utilizzando il linguaggio di markup  $\text{\LaTeX}_G$ , scelto poiché permette una facile separazione tra formattazione e presentazione. La scelta dell'editor da utilizzare è lasciata libera ai membri del gruppo.

**3.1.3.1.1 Comandi personalizzati** Sono stati definiti dei comandi  $\text{\LaTeX}$  personalizzati al fine di poter rendere più semplice ed immediata l'applicazione delle norme tipografiche. Questi comandi si occupano della corretta formattazione del testo secondo le norme che sono state definite.

## 3.2 Qualità

Nel documento *PianoDiQualifica.v.4.00*, in particolare nella sezione §2 Qualità di processo, devono essere descritte le tecniche utilizzate dal gruppo per garantire la qualità del prodotto:

- devono essere citati tutti gli standard che verranno seguiti per garantire la qualità dei processi e quindi la qualità del prodotto;
- devono essere presentati i processi individuati negli standard e ritenuti rilevanti.

Ogni processo dovrà essere specificato indicando:

- gli obiettivi derivanti da esso;
- le strategie ritenute utili per raggiungere gli obiettivi prefissati;
- le metriche che consentono una misurazione oggettiva e quantitativa di esso.

## 3.3 Configurazione

### 3.3.1 Norme

**3.3.1.1 Controllo di versione** Il controllo di versione di documenti e file sorgente viene fatto utilizzando il software  $\text{\_Git}_G$  e la piattaforma  $\text{\_GitHub}_G$ . Ad ogni modifica sostanziale ad un documento o ad un file sorgente impone che gli venga assegnato un nuovo numero di versione.

**3.3.1.2 Richieste di modifica** Ogni componente del gruppo può avanzare una richiesta di modifica al *Responsabile di progetto* seguendo queste norme:

1. la modifica deve essere presentata tramite la generazione di una issue, seguendo la sintassi riportata in §3.4.3.2;
2. il *Responsabile di progetto* ha il compito di analizzare tale richiesta e decidere se approvarla o meno;
3. In ogni caso, sia di accettazione della richiesta di modifica, sia in caso di rifiuto, il *Responsabile di progetto* è tenuto a motivare la scelta;
4. in caso di accettazione, il *Responsabile di progetto* deve assegnare il compito di realizzare tale modifica ad un membro del gruppo, in base al ruolo da lui ricoperto in quel momento;
5. una volta effettuata la modifica questa deve essere sottoposta a verifica e dev'essere fatta mantenendo traccia dello stato precedente.

### 3.3.1.3 Repository

**3.3.1.3.1 Struttura del repository** I file all'interno del repository<sub>g</sub> verranno organizzati secondo questa struttura:

- /LatexStyle;
  - /loghi;
- /RR;
  - /Esterni;
    - \* /VerbaliEsterni;
    - \* /AnalisiDeiRequisiti;
    - \* /Glossario;
    - \* /LetteraDiPresentazione;
    - \* /PianoDiProgetto;
    - \* /PianoDiQualifica;
    - \* /DefinizioneDiProdotto;
    - \* /ManualeUtente;
    - \* /ManualeUtenteEng;
    - \* /ManualeManutentore.

- /Interni;
  - \* /VerbalInterni;
  - \* /NormeDiProgetto;
  - \* /StudioDiFattibilità.

La struttura del repository andrà aggiornata nel corso del progetto con l'aggiunta di cartelle inerenti alle varie revisioni di avanzamento.

**3.3.1.3.2 Commit** L'esecuzione di ogni comando commit deve sottostare alle seguenti norme:

- Ad ogni commit è necessario specificare un messaggio nel quale si deve dare una descrizione, sintetica e il più possibile precisa, delle modifiche effettuate;
- Le modifiche apportate devono essere complete e testate con successo;
- È vietato l'utilizzo del comando `git add *` prima di un comando commit per evitare di includere nel repository file nascosti, temporanei o non voluti;
- È consigliato specificare il riassunto delle modifiche apportate ad ogni commit tramite il comando `git commit` rispetto al comando `git commit -m`, dove il messaggio deve essere scritto inline tra virgolette.

**3.3.1.3.3 Visibilità del repository** Per la condivisione e il versionamento dei configuration item è stato creato un repository privato su GitHub, raggiungibile all'indirizzo <https://github.com/esaraci/AtAVi>. L'accesso è consentito solo ai membri del gruppo.

#### 3.3.1.4 File

**3.3.1.4.1 Nomi dei file** I nomi dei file interni al repository devono sottostare alle seguenti norme:

- Devono contenere solo lettere, numeri, il carattere '\_', il segno meno ed il punto;
- Devono avere lunghezza minima di tre caratteri;
- Devono identificare in modo non ambiguo i file;

- Devono, nel caso contengano date, rispettare il formato YYYY-MM-DD.

è consigliato invece:

- Utilizzare la notazione CamelCase, invece di caratteri ‘\_’ e ‘-’ qualora il nome di un file fosse composto da più parole;
- Utilizzare nomi né troppo lunghi, né troppo corti, indicativamente tra i 10 e i 25 caratteri, estensione compresa;
- Specificare, quando possibile, l’estensione del file.

### 3.3.2 Procedure

**3.3.2.1 Richiesta di modifica** Il membro del gruppo che vuole effettuare una modifica deve presentare una richiesta formale al *Responsabile di progetto*. La richiesta deve avere i seguenti campi:

1. **Autore:** contiene nome, cognome e ruolo del richiedente della modifica;
2. **Documento:** contiene il nome del documento di cui si ritiene andrebbe fatta tale modifica;
3. **Urgenza:** indica una misura di quanto il richiedente ritiene necessario fare una modifica, può assumere solamente tali valori:
  - (a) “Alta”: in questo caso si ritiene che la modifica da fare sia importante, con pesanti conseguenze nell’organizzazione dello svolgimento delle attività future o che possa in qualche modo compromettere l’organizzazione delle attività future;
  - (b) “Media”: in questo caso il richiedente considera la modifica importante, ma non comporta forti conseguenze nell’organizzazione generale delle attività, ma può influenzare lo svolgimento di alcune di queste;
  - (c) “Bassa”: in questo caso la modifica proposta è di importanza secondaria, un suggerimento nell’effettuare alcune attività oppure che non influisce, o il suo peso è poco rilevante, nello svolgimento delle attività.
4. **Descrizione:** una descrizione dettagliata e motivata delle modifiche che si vogliono apportare;

5. **Decisione del responsabile di progetto:** questo campo va aggiunto successivamente alla decisione del *Responsabile di progetto*. Questo campo può assumere due valori:

- (a) “Approvata”;
- (b) “Respinta”.

In caso lo ritenga necessario, il *Responsabile di progetto* può aggiungere a questo campo la motivazione per cui ha approvato o meno la modifica.

**3.3.2.2 Aggiornamento del repository** Per l’aggiornamento del repository è prevista la seguente procedura:

- Dare il comando `git pull`. Nel caso in cui si verifichino dei conflitti:
  1. dare il comando `git stash` per accantonare momentaneamente le modifiche apportate;
  2. dare il comando `git pull`;
  3. dare il comando `git stash apply` per ripristinare le modifiche.

In questo modo il repository locale risulta aggiornato rispetto al repository remoto;

- dare il comando `git add NomeDelFile`, dove al posto di “NomeDelFile” si deve mettere il nome del file, o lista di nomi dei file, sul quale sono state effettuate delle modifiche;
- dare il comando `git commit`, e successivamente specificare sinteticamente il riassunto delle modifiche effettuate;
- dare il comando `git push`.

### 3.3.3 Strumenti

Lo strumento utilizzato per il versionamento di documenti e file sorgenti è Git. Nonostante siano presenti varie alternative, come Mercurial, Subversion e Bazaar, è stato scelto Git in quanto:

- soddisfa appieno le necessità di versionamento dei file per questo progetto;
- permette di versionare i file localmente, senza bisogno di una connessione internet attiva;



Come servizio di hosting per il repository è stato scelto, invece, GitHub. Entrambe le scelte sono state fatte anche perché più membri del gruppo avevano già avuto la possibilità di lavorare con tali servizi.

## 3.4 Verifica

### 3.4.1 Documenti

Il *Responsabile di progetto* ha il compito di dare inizio alla fase di verifica, assegnando i compiti ai verificatori. Questi ultimi, devono effettuare un'accurata verifica delle seguenti regole:

- deve essere utilizzata una sintassi corretta e il più possibile semplice;
- devono essere utilizzati periodi brevi;
- la struttura del documento deve essere semplice e intuitiva;
- devono essere rispettate le norme tipografiche menzionate precedentemente.

Durante l'intera attività di verifica i verificatori dovranno utilizzare il registro delle modifiche in modo tale da concentrare la loro attenzione sulle modifiche effettuate ad un documento e tenere traccia degli errori più comuni commessi nel redigere i documenti.

**3.4.1.1 Sintassi** I verificatori hanno il compito di identificare errori sintattici nei documenti. Ciò può essere fatto con l'ausilio di strumenti automatici, ma ogni documento deve essere sempre sottoposto a walkthrough in modo tale da individuare errori che non sono stati in grado di evidenziare. Inoltre hanno il compito di inserire nel glossario tutte quelle parole che possono essere fonte di ambiguità.

**3.4.1.2 Struttura del documento** I verificatori devono verificare che la struttura di ogni documento rispetti i contenuti del documento stesso.

**3.4.1.3 Resoconto attività di verifica sui documenti** La stesura del resoconto dell'attività di verifica dovrà essere fatta al termine dell'attività di verifica seguendo queste norme:

- qualora fossero stati trovati degli errori nelle attività di verifica precedenti, il numero di errori trovati andrà a sommarsi al numero di errori trovati nelle attività precedenti;

- in caso di errori ricorrenti riscontrati deve essere estesa la lista presentata in §4.3.5;
- il *Verificatore*, nel caso in cui dovesse trovare termini mancanti nel glossario, dovrà aggiornare la tabella definita nelle norme che regolano l'attività manuale di verifica.

**3.4.1.4 Attività manuale di verifica** Ogni attività manuale di verifica sui documenti deve essere accompagnata da un resoconto contenente il resoconto delle occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- periodi troppo lunghi o complessi;
- parole non appropriate;
- incongruenze;
- errori concettuali;
- violazioni di quanto stabilito nelle norme tipografiche.

Al fine di rendere più veloce e chiara l'analisi di queste voci, è consigliabile che questi dati vengano rappresentati in forma tabellare.

Inoltre, deve essere sempre presente una tabella contenente il numero di occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- termini candidati ad essere aggiunti al glossario;
- termini aggiunti al glossario.

**3.4.1.5 Attività automatica di verifica** Ogni attività automatica di verifica sui documenti deve essere accompagnata da un resoconto. Il resoconto deve contenere il numero di occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- errori ortografici;
- utilizzo errato L<sup>A</sup>T<sub>E</sub>X.

### 3.4.2 Diagrammi UML

I verificatori devono controllare che tutti i diagrammi UML prodotti rispettino lo standard UML e che siano corretti semanticamente.

**3.4.2.1 Diagrammi dei casi d'uso** Le verifiche che devono essere effettuate ai diagrammi dei casi d'uso devono riguardare:

- il rispetto dello standard UML;
- si deve verificare che la semantica del diagramma sia corretta, ovvero che il diagramma rappresenti effettivamente ciò che si vorrebbe descrivere e modellare.

### 3.4.3 Issue tracking

L'*issue<sub>g</sub>* tracking è un'attività di supporto per i *Verificatori*. In questo modo riescono a tenere traccia di potenziali errori e a segnalarli al *Responsabile di progetto*.

**3.4.3.1 Sintassi di una label** Le label sono delle etichette che è possibile associare ad ogni issue. Ogni etichetta rappresenta lo stato nel quale si trova la issue. Le uniche etichette accettate sono le seguenti:

- **Request:** rappresenta la richiesta di una issue;
- **Question:** identifica una issue della quale si sta discutendo per risolvere il problema;
- **ToDo:** identifica una issue per la quale è stata trovata una soluzione, ma che deve ancora essere applicata;
- **Working:** identifica una issue per la quale un membro del gruppo sta applicando la soluzione;
- **Rejected:** identifica una issue che non è stata accettata poiché risulta essere futile o dannosa;
- **Done:** identifica una issue che è stata completata.

**3.4.3.2 Sintassi di una Issue** Ogni issue avrà un nome che dovrà seguire la seguente notazione:

[D]:[S]

dove:

- **D** rappresenta la sigla del documento di interesse;
- **S** rappresenta la sintesi della descrizione del problema.

Segue, poi, la descrizione della issue che dovrà obbligatoriamente contenere:

- dove si trova il problema. Questo va specificato tramite sezione e, possibilmente, numero di riga dove inizia il problema;
- una descrizione dettagliata del problema, che deve comprendere:
  - errori associati: questi devono essere riportati nella forma di checkbox. Per fare ciò è necessario premettere alla descrizione di ogni errore i caratteri `- [ ]`. Questo permette di facilitare il tracciamento dello stato di avanzamento della issue;
  - eventuale porzione di codice che potrebbe essere causa del problema. Per fare ciò è consigliabile utilizzare il comando:

```
```nomelinguaggio  
codice da riportare  
```
```

Sostituendo “nomelinguaggio” al nome del linguaggio di programmazione utilizzato, si potrà vedere la porzione di codice evidenziata secondo la sintassi del linguaggio stesso, facilitandone la lettura.

- la motivazione per cui si è ritenuto necessario sollevare una issue.

**3.4.3.3 Gestione di una issue** Qualora i verificatori dovessero riscontrare delle anomalie, la procedura per la segnalazione e gestione del ticketing di una issue è la seguente:

1. il *Verificatore* che ha riscontrato un'anomalia<sub>c</sub> dovrà aprire una nuova issue assegnandole l'etichetta Request;
2. il *Responsabile di progetto* dovrà valutare la issue e decidere se:
  - la issue risulta essere futile o dannosa. In questo caso verrà rimossa l'etichetta Request ed assegnata l'etichetta Rejected. La issue verrà poi chiusa, terminando la procedura.
  - la issue risulta essere appropriata. In questo caso verrà rimossa l'etichetta Request ed assegnata l'etichetta Question.
3. i membri del gruppo dovranno discutere e proporre nuove idee al fine di risolvere la issue. Questa discussione non dovrà protrarsi per più di tre giorni. Qualora non si dovesse essere trovata una soluzione ottimale, il *Responsabile di progetto* dovrà decidere quale sia la soluzione migliore.

4. il *Responsabile di progetto*, trovata la soluzione da applicare, assegnerà, al membro del gruppo da lui ritenuto opportuno, un ticket<sub>g</sub>. L'assegnatario del ticket è la persona incaricata all'applicazione della soluzione relativa alla issue. Inoltre, il *Responsabile di progetto* dovrà rimuovere dalla issue l'etichetta Question ed assegnare l'etichetta ToDo.
5. quando l'incaricato inizierà a lavorare sulla issue rimuoverà l'etichetta ToDo ed assegnerà l'etichetta Working. Ad ogni nuovo commit, l'incaricato dovrà riferire, nel messaggio del commit stesso, la issue sulla quale sta lavorando in modo da tener traccia delle modifiche effettuate.
6. una volta che l'incaricato avrà terminato le modifiche, queste dovranno essere caricate nel repository e l'etichetta della issue dovrà essere modificata da Working a Done. Dopo che sono state svolte queste operazioni, la issue verrà chiusa.

**3.4.3.4 Strumento per l'issue tracking** Lo strumento scelto per la gestione di questa attività è GitHub e il suo servizio Issues. Questo strumento è utile anche al *Responsabile di progetto* per assegnare i compiti di correzione degli errori.

#### 3.4.4 Tecniche di analisi

Per poter verificare la qualità del prodotto<sub>g</sub> è stato scelto di applicare delle tecniche di analisi sui documenti e sul codice.

**3.4.4.1 Analisi statica** Per analisi statica si intende la valutazione di un sistema basata su struttura, contenuto e documentazione senza che questo venga eseguito. Questa tecnica è applicabile sia al codice, che alla documentazione stessa. L'analisi statica può avvenire con due modalità:

- **Walkthrough:** questa tecnica deve essere applicata quando non si sa che errori o problematiche si stanno cercando. La tecnica consiste nel leggere il codice sorgente o il documento da cima in fondo per trovare anomalie di qualsiasi tipo;
- **Inspection:** questa tecnica deve essere applicata quando si ha idea della problematica che si sta cercando e consiste in una lettura mirata del documento/codice, sulla base di una lista degli errori precedentemente stilata e presentata in sezione §4.3.5.

La tecnica *walkthrough* è molto onerosa e deve essere utilizzata:

- nelle prime fasi del progetto;
- quando non è già presente una lista degli errori comuni;
- quando non si è ancora sufficientemente preparati riguardo un aspetto nuovo del progetto.

Avanzando nelle fasi del progetto sarà utile stilare una lista quanto più possibile completa di errori comuni, in modo tale da evitare la tecnica *walkthrough* e applicare l'*inspection*.

**3.4.4.2 Analisi dinamica** L'analisi dinamica è una forma di valutazione di un sistema software, oppure di qualche suo componente, basato sull'osservazione del suo comportamento durante l'esecuzione. Questa tecnica non è applicabile, quindi, per trovare errori nella documentazione. I test che devono essere implementati devono essere in numero relativamente ridotto e il più possibile di valore dimostrativo.

### 3.4.5 Test

**3.4.5.1 Classificazione dei test** È compito dei *Programmatori* creare test di unità e integrazione e compito degli *Analisti* creare quelli di sistema e accettazione. I test hanno il fine di verificare che tutti i requisiti individuati dagli *Analisti* siano soddisfatti e che le componenti progettate durante la definizione dell'architettura ad alto livello funzionino correttamente. I test di integrità e di unità devono essere codificati come segue:

$$T[X][Y]$$

dove:

1. **X** rappresenta il tipo di test. Può assumere solo i seguenti valori:

- **I**: indica un test di integrazione;
- **U**: indica un test di unità.

2. **Y** rappresenta il codice univoco incrementale per ogni test.

I test di sistema e di accettazione devono essere codificati come segue:

$$T[X][Y][Z][W]$$

dove:

1. **X** rappresenta il tipo di test. Può assumere solo il seguente valori:

- **A**: indica un test di accettazione;
  - **S**: indica un test di unità.
2. **Y** indica la tipologia del requisito. Deve assumere solo i seguenti valori:
- **F**: indica un requisito funzionale;
  - **Q**: indica un requisito di qualità;
  - **P**: indica un requisito prestazionale;
  - **V**: indica un requisito vincolo.
3. **Z** indica l'importanza strategica del requisito. Deve assumere solo i seguenti valori:
- **Obb**: indica un requisito obbligatorio;
  - **Des**: indica un requisito desiderabile;
  - **Opz**: indica un requisito opzionale.
4. **W** rappresenta il codice univoco di ogni requisito in forma gerarchica.

### 3.5 Validazione

Il processo di validazione ha l'obiettivo di verificare che:

- il prodotto finale sia conforme a quanto è stato pianificato;
- il prodotto finale sia abile nell'isolare e minimizzare gli effetti degli errori.

#### 3.5.1 Responsabilità

Di seguito si elencano le rispettive responsabilità per la validazione del prodotto:

- i *Verificatori* hanno il compito di eseguire con attenzione i test, tracciandone i risultati che andranno poi valutati;
- il *Responsabile di progetto* revisiona i risultati dei test e decide se accettarli o ripeterli. Ha inoltre il compito di informare il committente dell'esecuzione e risultato dei test, fornendo indicazioni sulla possibilità di eseguirli in modo indipendente.

### 3.5.2 Procedure

**3.5.2.1 Procedura per la validazione** I passi per eseguire l'attività di validazione sono i seguenti:

1. i *Verificatori* eseguono manualmente i test sul prodotto finale, tracciando accuratamente i risultati ottenuti;
2. il *Responsabile di progetto* analizza i risultati e decide se:
  - (a) accettarli;
  - (b) chiedere la ripetizione di alcuni o tutti i test, eventualmente con *Verificatori* diversi.
3. una volta accettati i test, il *Responsabile di progetto* consegna i risultati al proponente, informandolo sulle modalità di esecuzione indipendente della validazione.



## 4 Processi organizzativi

### 4.1 Norme

#### 4.1.1 Comunicazioni

**4.1.1.1 Interne** Per le comunicazioni interne è stato scelto di utilizzare lo strumento Messages messo a disposizione da *TeamWork<sub>G</sub>*. La procedura per inviare un messaggio ai membri del gruppo è la seguente:

1. accedere a *TeamWork* ed entrare nella sezione Messages;
2. creare un nuovo messaggio inserendo oggetto e corpo del messaggio;
3. selezionare i destinatari del messaggio ed inviare.

Per rispondere ad un messaggio ricevuto la procedura è la seguente:

1. accedere a *TeamWork* ed entrare nella sezione Messages;
2. selezionare il messaggio al quale si vuole rispondere ed inviare la risposta.

Per comunicazioni brevi e tempestive è stato scelto l'utilizzo dello strumento di messaggistica *Telegram<sub>G</sub>*.

Nel momento in cui i sistemi di comunicazione portassero a decisioni rilevanti riguardo lo sviluppo del progetto, dovrà essere stilato un verbale. I membri del gruppo sono tenuti a prestare attenzione al numero di messaggi diffusi per non creare difficoltà di comunicazione.

**4.1.1.2 Esterne** Per la comunicazioni esterne è stato creato un apposito indirizzo di posta elettronica: [answer.group17@gmail.com](mailto:answer.group17@gmail.com). Questa mail è accessibile a tutti i membri del gruppo in modo tale che tutti possano essere a conoscenza delle discussioni avvenute con componenti esterne. È stato creato un account *Slack* in accordo con il proponente per favorire un'interazione più tempestiva e informale.

**4.1.1.3 Composizione delle email** Questa sezione tratta le norme da rispettare nella composizione delle email, sia per la comunicazione interna che per quella esterna.

**4.1.1.3.1 Mittente** Le norme che devono essere rispettate sono le seguenti:

- nel caso di comunicazione interna via email il mittente deve essere l'indirizzo di posta elettronica personale del membro del gruppo che ha scritto l'email;
- in caso di comunicazione esterna l'unico indirizzo che deve essere utilizzato è `answer.group17@gmail.com`;
- qualora la comunicazione debba avvenire tra un numero ristretto di persone all'interno del gruppo, questi potranno utilizzare i loro indirizzi email personali.

**4.1.1.3.2 Destinatario** Le norme che devono essere rispettate sono le seguenti:

- nel caso di comunicazione interna al gruppo l'unico destinatario deve essere l'indirizzo mail del gruppo per facilitare il ricevimento di tale mail a tutti i membri del gruppo;
- in caso di comunicazione esterna i destinatari possono essere il Prof. Tullio Vardanega, in qualità di committente, il Prof. Riccardo Cardin oppure i proponenti del progetto, a seconda dello scopo dell'email.

**4.1.1.3.3 Oggetto** Le norme che devono essere rispettate sono le seguenti:

- l'oggetto dell'email deve sintetizzare il contenuto dell'email stessa in modo più chiaro ed esaustivo possibile;
- l'oggetto di una nuova email deve essere differente rispetto alle email inviate e ricevute in precedenza, in modo tale da rendere facilmente identificabile ogni messaggio;
- per comporre un messaggio di risposta è necessario anteporre all'oggetto il prefisso "Re:";
- nel caso di inoltro di un messaggio è obbligatorio aggiungere il prefisso "I:".

**4.1.1.3.4 Corpo** Il corpo del messaggio deve essere esaustivo, sintetico e deve essere comprensibile a tutti i destinatari del messaggio. In caso di risposta o inoltro è preferibile aggiungere la nuova parte di testo all'inizio dell'email per permettere una più facile lettura del contenuto.

**4.1.1.3.5 Allegati** Si consiglia di non fare uso di allegati per lo scambio di documenti o file, a meno che non sia strettamente necessario. È preferibile, invece, caricare questi file in una cartella di Google Drive<sub>g</sub> e inviare per email il link al documento o file desiderato. Sia in caso di invio di allegati, che di link a documenti o file è buona norma inserire una breve descrizione di presentazione dell'allegato in modo tale che sia possibile capire di cosa si tratta.

## 4.1.2 Riunioni

Il *Responsabile di progetto* ha il compito di indire le riunioni, sia interne che esterne, inviando un'email di convocazione strutturata in questo modo:

- **oggetto:** convocazione della riunione N per il giorno YYYY-mm-DD;
- **corpo:**
  - Data;
  - Ora;
  - Luogo;
  - Tipo di riunione;
  - Ordine del giorno.

Dove N rappresenta il numero della riunione e tipo indica se la riunione sia interna richiesta dal *Responsabile di progetto*, interna richiesta da uno o più membri del gruppo oppure esterna. Le informazioni sulle riunioni devono essere presentate almeno tre giorni prima in modo tale che i membri del gruppo possano organizzare i loro impegni ed essere presenti alla riunione.

### 4.1.2.1 Interne

**4.1.2.1.1 Convocazione** In generale, il compito di convocare le riunioni interne spetta al *Responsabile di progetto*, che può indirle quando più lo ritiene opportuno. Gli altri componenti del gruppo possono richiedere una riunione interna straordinaria, presentando al *Responsabile di progetto* le motivazioni per le quali si ritiene necessaria una riunione. In questi casi il *Responsabile di progetto* può:

- autorizzare lo svolgimento della riunione;
- negare lo svolgimento della riunione, nel caso in cui non ritenga le motivazioni presentate valide per richiedere una riunione;

- suggerire mezzi di comunicazione differenti.

In ogni caso spetta al *Responsabile di progetto* decidere data, ora e luogo dell'incontro contattando i membri del gruppo e chiedendo loro la disponibilità; questi sono tenuti a rispondere tempestivamente per favorire l'organizzazione.

**4.1.2.1.2 Gestione** Le norme che devono essere rispettate sono le seguenti:

1. all'inizio di ogni riunione interna il *Responsabile di progetto* deve nominare un Segretario;
2. il Segretario che ha il compito di redigere la minuta della riunione, catturando possibilmente tutti e soli gli aspetti più importanti della riunione stessa;
3. terminato l'incontro, il Segretario ha il compito di redigere il verbale dell'incontro;
4. il verbale deve essere archiviato nel repository del gruppo, per essere visibile a tutti i membri.

Durante le riunioni i partecipanti devono tenere un comportamento che favorisca la discussione all'interno del gruppo e la trattazione di tutti gli argomenti previsti.

#### **4.1.2.2 Esterne**

**4.1.2.2.1 Convocazione** Il *Responsabile di progetto* ha questi compiti:

- fissare le riunioni esterne con i Proponenti o con i Committenti, contattandoli tramite la casella di posta elettronica del gruppo;
- accordarsi con i Proponenti o Committenti riguardo data, orario e luogo dell'incontro, tenendo conto anche della disponibilità dei membri del gruppo e cercando, per quanto possibile, di far partecipare tutti.

**4.1.2.2.2 Gestione** Le norme che devono essere rispettate sono le seguenti:

1. viene nominato un Segretario che ha il compito di scrivere un verbale;
2. il *Responsabile di progetto* deve redigere il verbale della riunione esterna, avvalendosi dell'aiuto di tutti i membri del gruppo presenti all'incontro, per avere una trascrizione più fedele possibile dei contenuti.

**4.1.2.3 Verbale** Il verbale di una riunione è un documento nel quale vengono riassunti gli argomenti trattati e indicate eventuali decisioni prese nel corso della riunione stessa.

Il compito della stesura del verbale spetta ad un Segretario nominato all'inizio di una riunione dal *Responsabile di progetto*. I verbali dovranno avere la struttura presentata nella sezione §3.1.1.3.4.3.

### 4.1.3 Ruoli di progetto

Per lo sviluppo collaborativo del progetto, ai membri del gruppo saranno assegnati dei ruoli differenti che corrispondono a figure professionali in modo da seguire queste norme:

- ogni membro deve ricoprire ogni ruolo almeno una volta;
- deve essere garantito che il ruolo di ciascun membro del gruppo non sia in conflitto con il ruolo che ha ricoperto in passato.

Il *Verificatore* deve controllare che siano rispettate queste condizioni, in caso contrario dovrà avvertire il *Responsabile di progetto* che dovrà risolvere il problema.

**4.1.3.1 Responsabile di Progetto** Il *Responsabile di progetto* rappresenta il gruppo e fa da tramite con il fornitore e con il committente, accentrando su di sé la responsabilità di scelta e approvazione. Per questo motivo ha responsabilità su:

- **Pianificazione**, coordinamento e controllo delle attività;
- **Gestione delle risorse**;
- **Analisi** e gestione dei **rischi**;
- **Approvazione** dei documenti;
- **Approvazione** dell'offerta economica;
- **Convocazione** delle riunioni interne;
- **Relazioni** esterne;
- **Assegnazione** delle attività a persone.

Per questi motivi ha il compito di:

- assicurarsi che le attività di verifica e validazione siano svolte seguendo le *NormeDiProgetto\_v.4.00*;
- garantire il rispetto dei ruoli e dei compiti assegnati nel *PianoDiProgetto\_v.4.00*.

**4.1.3.2 Amministratore** L'*Amministratore* è responsabile del controllo e della gestione dell'ambiente di lavoro. In particolare deve preoccuparsi di:

- **Equipaggiare** l'ambiente di lavoro con strumenti, procedure, infrastrutture e servizi a supporto dei processi che permettano di automatizzare il più possibile le attività o parti di esse;
- **Garantire** che l'ambiente di lavoro sia completo, dotato di tutti gli strumenti necessari al progetto, ordinato e aggiornato;
- **Controllare** le versioni e configurazioni del prodotto;
- **Gestire** la documentazione, controllarne la diffusione, la disponibilità e l'archiviazione;
- **Fornire** procedure e strumenti per il monitoraggio e segnalazione per il controllo qualità;
- **Risolvere** problemi legati alla gestione dei processi tramite l'adozione di strumenti adatti.

L'*Amministratore* redige inoltre le *Norme di progetto*, dove viene spiegato l'utilizzo degli strumenti, e deve redigere la sezione del *Piano di qualifica* dove vengono descritti gli strumenti e i metodi atti alla verifica. L'*Amministratore* non compie scelte gestionali, ma tecnologiche concordate con il *Responsabile di progetto*.

**4.1.3.3 Analista** L'*Analista* è il responsabile delle attività di analisi. Le mansioni che gli competono riguardano:

- **Comprendere** la natura e la complessità del problema tramite l'analisi dei bisogni e delle fonti;
- **Classificare** i requisiti;
- **Stendere** i diagrammi dei casi d'uso;
- **Assegnare** i requisiti a parti distinte del sistema;

- **Assicurarsi** che i requisiti trovati siano conformi alle richieste del committente.

L'*Analista* non si occupa di trovare una soluzione al problema, ma lo definisce redigendo lo *Studio di fattibilità* e l'*Analisi dei requisiti*. Partecipa alla definizione del *Piano di qualifica* in quanto conosce a fondo il problema e deve avere chiari i livelli di qualità richiesti, oltre alle procedure per ottenerli.

**4.1.3.4 Progettista** Il *Progettista* è il responsabile delle attività di progettazione. I compiti a lui affidati comprendono:

- produrre una soluzione attuabile e che sia comprensibile e soddisfacente per gli stakeholders;
- fare in modo che le soluzioni applicate al prodotto siano note ed ottimizzate;
- fare in modo che il prodotto sia facilmente manutenibile;
- organizzare la costituzione del prodotto in modo che sia realizzabile con costi e scadenze prefissate.

Il *Progettista* redige i documenti del documento *Definizione di prodotto* e si occupa delle sezioni del *Piano di qualifica* relative alle metriche di verifica della programmazione.

**4.1.3.5 Programmatore** Il programmatore è responsabile delle attività di codifica e delle componenti di ausilio necessarie per l'esecuzione delle prove di verifica e validazione. In particolare ha i seguenti compiti:

- implementare in maniera rigorosa le soluzioni descritte dal *Progettista*;
- scrivere codice che sia documentato, manutenibile e che rispetti le metriche stabilite per la scrittura del codice;
- realizzare i test per la verifica e la validazione del codice stesso.

Il programmatore, infine, deve occuparsi di redigere il *Manuale utente*.

**4.1.3.6 Verificatore** Il *Verificatore* è il responsabile delle attività di verifica. Le mansioni che gli competono sono:

- garantire che l'attuazione delle attività sia conforme alle norme stabilite;
- verificare che le attività svolte non abbiano introdotto errori;
- controllare la conformità di ogni stadio del ciclo di vita del prodotto.

Il *Verificatore* deve redigere le sezioni del *Piano di qualifica* riguardanti l'esito e la completezza delle verifiche effettuate.

**4.1.3.7 Rotazione dei ruoli** Ogni membro del gruppo dovrà ricoprire ciascuno dei ruoli del progetto. La pianificazione dovrà essere redatta prestando attenzione a quanto segue:

- ogni membro del gruppo non dovrà mai ricoprire un ruolo che preveda la verifica dell'operato svolto da lui in precedenza poiché questo potrebbe portare ad un conflitto di interesse;
- bisogna tener conto dei possibili impegni o interessi dei singoli membri del gruppo;
- ciascun membro dovrà assicurare l'esclusivo svolgimento del ruolo a lui assegnato.

#### 4.1.4 Ticketing

I ticket<sub>c</sub> rappresentano l'associazione tra un membro del gruppo ed una attività. Lo strumento scelto per la loro creazione e gestione è Teamwork. Per ogni ticket valgono le seguenti norme:

- possono essere creati solamente dal *Responsabile di progetto*;
- alla creazione di un nuovo ticket, l'assegnatario deve esserne informato;
- è consigliabile che non vi siano più di 3 ticket per ogni attività. Nel caso in cui questo dovesse accadere, il *Responsabile di progetto* dovrebbe valutare la possibilità di suddividere l'attività in sotto-attività più piccole, ciascuna delle quali realizzabile da una sola persona.

È consigliabile che i membri del gruppo accedano quotidianamente a Teamwork per verificare quali siano i ticket a loro assegnati e riportare lo stato di avanzamento dell'attività a loro assegnata.



## 4.2 Procedure

### 4.2.1 Creazione e gestione di un ticket

Per la creazione di un nuovo ticket il *Responsabile di progetto*, deve accedere a Teamwork ed entrare nella sezione Tasks. Se il ticket deve essere associato ad una nuova attività allora la procedura da seguire è la seguente:

1. creare un nuovo task<sub>c</sub> impostando i parametri desiderati;
2. selezionare i membri del gruppo a cui associare il nuovo task;
3. confermare la creazione del nuovo task.

Se il ticket, invece, deve essere associato ad un'attività già presente allora la procedura da seguire è la seguente:

1. selezionare il task a cui associare i nuovi ticket;
2. modificare il task e selezionare i membri del gruppo a cui associare il task;
3. confermare le modifiche effettuate.

Al termine del punto 3 degli elenchi soprastanti, Teamwork provvederà ad inviare in automatico una email di notifica alle persone assegnatarie del task. Per riportare lo stato di avanzamento dell'attività, gli assegnatari del ticket dovranno:

1. accedere a Teamwork ed entrare nella sezione task;
2. selezionare il task di cui si vuole riportare l'avanzamento;
3. aggiornare la percentuale di avanzamento del task selezionato.

Al termine dell'attività gli assegnatari dovranno ripetere la procedura sopra riportata ed infine selezionare il checkbox indicante il completamento dell'attività. Questo comporterà la chiusura del ticket.

## 4.3 Strumenti

### 4.3.1 Pianificazione

Lo strumento scelto per la gestione delle attività di pianificazione di progetto è Teamwork. Le caratteristiche rilevanti di questo software sono le seguenti:

- creazione di attività e sotto-attività assegnabili ad uno o più membri del progetto;
- possibilità di creare dipendenze tra le attività;
- possibilità di assegnare priorità differenti ad ogni attività;
- creazione di milestones<sub>G</sub> da impostare sul calendario;
- creazione automatica di diagrammi e report, tra cui il diagramma di Gantt<sub>G</sub>;
- dashboard<sub>G</sub> che permette di aver un riepilogo dello stato di avanzamento del progetto, con segnalazione di eventuali ritardi;
- strumento per la segnalazione dei rischi;
- sistema di gestione delle notifiche per ogni attività svolta.

#### 4.3.2 Tracciamento dei requisiti, test, casi d'uso

Lo strumento scelto per il tracciamento dei requisiti, dei test e dei casi d'uso è Trender<sub>G</sub> un software open source<sub>G</sub> che permette, attraverso un' accattivante interfaccia grafica, il tracciamento dei requisiti, dei test e dei casi d'uso. Vengono elencate di seguito le funzionalità offerte da Trender che verranno maggiormente utilizzate da *Answer*:

- gestione dei requisiti;
- gestione dei test;
- gestione dei casi d'uso;
- gestione degli attori;
- stampa e backup dei dati presenti.

#### 4.3.3 Calcolo del consuntivo

Per il calcolo del consuntivo vengono utilizzate due funzionalità offerte da *TeamWork*:

1. la rendicontazione delle ore di lavoro da parte di ogni membro del gruppo;

2. l'esportazione delle ore rendicontate tramite un foglio di calcolo in formato Microsoft Excel.

Gli strumenti utilizzati dal *Responsabile di progetto* per la stesura del consuntivo sono i seguenti:

- *Team Work*;
- Microsoft Excel.

#### 4.3.4 Meccanismi di controllo

Sono stati adottati dei meccanismi, nell'ambiente creato, per:

- controllare l'andamento delle attività;
- permettere un aggiornamento facilitato della pianificazione;
- rendicontare le ore di lavoro spese nelle varie attività.

**4.3.4.0.1 Controllo metriche di progetto** L'introduzione delle metriche consente di quantificare in modo più obiettivo possibile le performance del gruppo nello svolgimento del progetto attraverso la misurazione dell'insieme di indicatori che ne fanno parte. Tipicamente uno degli usi più importanti delle metriche è quello di misurare l'avanzamento del progetto a fronte del piano. Il loro utilizzo consente di:

- identificare i problemi di costo/schedulazione prima che diventino criticità;
- aiutare il team a focalizzarsi sul completamento delle proprie attività.

In particolare le metriche Budget Variance(BV) e Schedule Variance(SV) permettono rispettivamente di:

- indicare se si è speso di più o di meno rispetto a quanto previsto;
- indicare se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di progetto pianificate nella baseline.

I valori aggiornati di tali metriche sono riportati nel documento *PianoDiQualifica\_v.4.00*.

**4.3.4.0.2 Meccanismi di rendicontazione** Il sistema di ticketing adottato mette a disposizione la rendicontazione delle ore di lavoro. Tale sistema permette di visualizzare le ore di lavoro in base all'attività svolta.

#### 4.3.5 Lista di controllo

Durante l'applicazione della tecnica del Walkthrough ai documenti sono stati riportati più frequentemente i seguenti errori:

- norme stilistiche:
  - la prima parola di una voce dell'elenco puntato inizia con una lettera maiuscola;
  - la voce dell'elenco puntato termina con un punto anziché con un punto e virgola o viceversa;
  - i due punti ad inizio elenco puntato risultano in grassetto;
  - errori di battitura;
  - le immagini e il testo devono stare dentro il range della pagina.
- LATEX:
  - mancato utilizzo dei comandi LATEX personalizzati;
  - spaziatura errata dopo un comando personalizzato `\LaTeX`.
- casi d'uso:
  - mancato rispetto del template stabilito per i punti trattati nei casi d'uso.
- glossario:
  - sono stati evidenziati dei termini che non andavano nel Glossario;
  - non sono stati evidenziati dei termini che sono presenti nel Glossario;
  - evidenziate con la marcatura  $\_g$  tutte le occorrenze di un termine nel documento.

### 4.4 Formazione dei membri del gruppo

Per lo svolgimento del progetto didattico ogni membro del gruppo necessita di tempo dedicato alla formazione riguardo gli ambiti e le tecnologie impiegati nel progetto. Solo parte di queste ore saranno a carico del Proponente.