

1 Processi primari

1.1 Fornitura

1.1.1 Studio di fattibilità

Il documento riguardante lo studio di fattibilità deve essere redatto rapidamente ed in modo accurato dagli *Analisti* sulla base di ciò che è emerso nelle prime riunioni, nelle quali si deve discutere di temi riguardanti i capitolati, come:

- rischi nell'affrontare ogni capitolato_g;
- rapporto tra i costi ed i benefici, sia in base al mercato attuale che futuro, sia in base al costo di produzione e alla possibile redditività futura;
- il dominio applicativo e tecnologico di ogni capitolato_g.

1.2 Sviluppo

1.2.1 Analisi dei requisiti

L'*Analisi dei requisiti* è il documento dove devono essere catalogati e descritti tutti i requisiti che il prodotto_g finale deve soddisfare. Ogni requisito deve emergere da una delle seguenti fonti:

- capitolati_g d'appalto;
- incontri con il proponente;
- incontri con il committente;
- valutazioni effettuate durante riunioni interne al gruppo.

Tale documento deve inoltre riportare il modo in cui ogni requisito deve essere verificato.

1.2.1.1 Classificazione dei casi d'uso È compito degli *Analisti* redigere una descrizione, dare una classificazione e fornire un diagramma conforme allo standard UML_g per ogni caso d'uso. Ogni caso d'uso dev'essere descritto con le seguenti informazioni, possibilmente in quest'ordine:

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

UC[X].[Y]

dove:

- **X** è il codice univoco del padre;
- **Y** è un codice progressivo di livello.

Il codice progressivo può includere diversi livelli di gerarchia separati da un punto.

2. titolo, che deve descrivere sinteticamente il caso d'uso;
3. attori principali;
4. attori secondari, se questi sono presenti;
5. precondizioni, ovvero le condizioni che necessariamente devono verificarsi prima del caso d'uso;
6. postcondizioni, ciò che deve essere verificato 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. Scenari 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.

1.2.1.2 Classificazione dei requisiti È compito degli *Analisti* redigere e classificare i requisiti. I requisiti devono essere classificati in base al tipo e alla priorità, utilizzando la seguente notazione:

R[X][Y][Z]

dove:

1. **X** 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.
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** rappresenta il codice univoco di ogni requisito in forma gerarchica.

1.2.2 Progettazione

Norme, procedure e strumenti riguardanti la progettazione verranno definiti nelle versioni successive di questo documento.

1.2.3 Codifica

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

I *Programmatori*, dovendo sviluppare un'applicazione Android, si dovranno seguire le indicazioni fornite dalla guida Google Java Style¹.

1.2.3.2 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.

1.2.3.3 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.

1.2.3.4 Documentazione

¹<https://google.github.io/styleguide/javaguide.html>

1.2.3.4.1 File I file contenenti codice dovranno avere un'intestazione contenente:

```
/**
 * @author Nome dell'autore
 * @version Versione corrente del file
 * @since Versione del file nel momento
 *       dell'aggiunta al progetto
 *
 * Descrizione del file
 */
```

1.2.3.4.2 Classi Prima di ogni classe dovrà essere presente un commento, strutturato come l'intestazione dei file.

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

```
/**
 * Descrizione del metodo
 * @param NomeParametro1 Descrizione del primo
 *       parametro
 * ...
 * @param NomeParametroN Descrizione del N-esimo
 *       parametro
 * @return TipoDiRitorno Valore ritornato dal
 *       metodo (opzionale)
 * @throws TipoDiEccezione Motivo di lancio
 *       dell'eccezione
 */
```

Per i metodi senza valore di ritorno (con tipo di ritorno void e costruttori) la voce `@return` deve essere omessa.

1.2.3.4.4 Casi eccezionali

1.2.4 Strumenti

1.2.4.1 Strumento per la creazione dei diagrammi UML Lo strumento per la creazione dei diagrammi UML_g utilizzato è Astah_g.

1.2.4.2 Strumento per il tracciamento dei requisiti Lo strumento scelto per la il tracciamento dei requisiti è Tracy_g. Questo software_g è stato sviluppato dal gruppo di Ingegneria del Software Don't Panic. Il software_g è stato scelto per le seguenti caratteristiche:

- open source_g;
- tracciamento dei requisiti;
- tracciamento use case;
- tracciamento delle fonti;
- stesura automatica in L^AT_EX_g dei requisiti.

Poiché questo software_g non risulta essere completamente perfetto, il gruppo ha previsto di riadattarlo sulla base delle esigenze che sono emerse durante la stesura dell'*Analisi dei requisiti*.