

La struttura **URD** (User Requirements Document) o **DRU** (Documento Requisiti Utente) (altre volte chiamato **SRS** - System Requirements Specification) è la seguente:

- * **Introduzione**
- * **Glossario**
- * **User Requirements Definition**
- * **System Requirements**
- * **System Architectural Models**
- * **Process and Evolution**
- * **Appendici**

Così definiti:

- * **Introduzione:** Introduzione al documento con una sezione dedicata alle specifiche di massima del sistema (2/3 pagine che descrivono il progetto)

In questa sezione si dovrà descrivere il progetto utilizzando il testo fornito e i chiarimenti svolti a lezione. Si dovrà utilizzare linguaggio naturale e, ove necessario, uno schema di massima per chiarire gli elementi salienti del sistema in oggetto.

- * **Glossario:** Definizione della terminologia di contesto e specifica al documento

Definire in una tabella tutti i termini e gli acronimi che verranno utilizzati nel documento allo scopo di fornire al lettore una base per la lettura e la codifica della terminologia utilizzata.

- * **User Requirements:** Sezione dedicata alla descrizione in linguaggio naturale con l'ausilio di use cases e templates i requisiti utente.

In questa sezione si dovranno descrivere i requisiti utente del sistema attraverso l'utilizzo di use cases UML e tabelle di supporto (con identificativo, descrizione, condizioni, etc) in modo da chiarire i requisiti utente, gli attori del sistema e le interazioni con il sistema. Definite una volta per tutte un template per la descrizione degli use case e usatelo sempre. Tale template deve includere in ogni caso le seguenti voci: (a) Nome univoco dello Use case, (b) Breve descrizione (c) Attori (d) precondizioni (e) scenario principale (f) (principali) scenari alternativi (g) postcondizioni. Potete aggiungere altri elementi che ritenete importanti.

Prendete in considerazione anche i seguenti punti:

- 1) gli use case vi aiutano per i requisiti funzionali, dovete però anche descrivere dettagliatamente i requisiti non funzionali (ad esempio requisiti di safety);
- 2) per quanto riguarda i requisiti non funzionali di performance (ad esempio tempi di risposta) potete collegarli, quando possibile, agli use case;
- 3) per ogni requisito non funzionale specificare come verrà testato e verificato.

- * **System Requirements:** Specifica dei requisiti funzionali, non funzionali e di dominio

In questa sezione si dovranno descrivere i requisiti di sistema in forma di tabella, indicando il loro identificativo univoco, la descrizione ed altri campi atti a definire il requisito (come l'origine, la priorità rispetto all'implementazione, etc.). Si ricorda che questa sezione non fa uso di diagrammi UML e che i requisiti di sistema dovranno essere utilizzati per supportare le fasi di ingegneria del software anche attraverso matrici di tracciabilità rispetto all'implementazione, ai test, etc. (le matrici sono opzionali).

- * **System Architectural Models:** Modelli UML 2.0 relativi al sistema. In particolare si deve fare uso dei diagrammi strutturali e comportamentali più comuni (come ad esempio sequenze, collaboration, class, objects, state, ...).

In questa sezione si dovrà dare un modello del sistema il più possibile completo e generale. Utilizzare il numero e le tipologie di diagrammi che più rispondono alle caratteristiche del sistema stesso.

Provate a mettere in corrispondenza i componenti architetturali individuati con i requisiti individuati nelle fasi precedenti, indicando quale componente realizza quali requisiti (tracciamento dei requisiti).

Utilizzate il linguaggio OCL per integrare i diagrammi con statement che incorporano requisiti non funzionali e vincoli sul funzionamento del sistema.

Opzionali sono l'utilizzo dei diagrammi fisici (component e deployment)

- * **Process and Evolution:** Cenni al processo di ingegneria del software scelto per implementare correttamente la specifica, il disegno, l'integrazione, il test ed il deployment di sistema. Opzionale anche un accenno all'evoluzione ed al mantenimento del sistema.

- * **Appendici:** Da usare nel caso si vogliano dare dei dettagli di parti del documento e anche nel caso si vogliano descrivere le interfacce utente, snapshot del sistema, codice, etc.

Appendice su Ambiente di Sviluppo In questa Appendice discutete l'ambiente di sviluppo (IDE) che intendereste adottare per la realizzazione del sistema, motivando le vostre scelte. È opportuno prendere in considerazione ambienti di sviluppo che siano integrati con la fase di modellazione del sistema. Rispetto all'IDE dai voi scelto discutete (sinteticamente) almeno i seguenti punti:

1. integrazione con i modelli del sistema e le specifiche dei requisiti;
2. supporto allo sviluppo e al lavoro in team con riferimento al debugging, al building e alla gestione delle versioni;
3. possibilità di generare documentazione in modo automatico.

Note generali:

- L'implementazione del sistema non è richiesta.
- Utilizzare un tool per la modellazione come Visual Paradigm o equivalente.