Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci

Quality Management Plan

MyBomber

Riferimento	
Versione	1.0
Data	14/01/2022
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Gaetano Mauro
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
28/12/2021	0.1	Prima stesura	Gaetano Mauro
14/01/2022	1.0	Revisione	Gaetano Mauro

Sommario

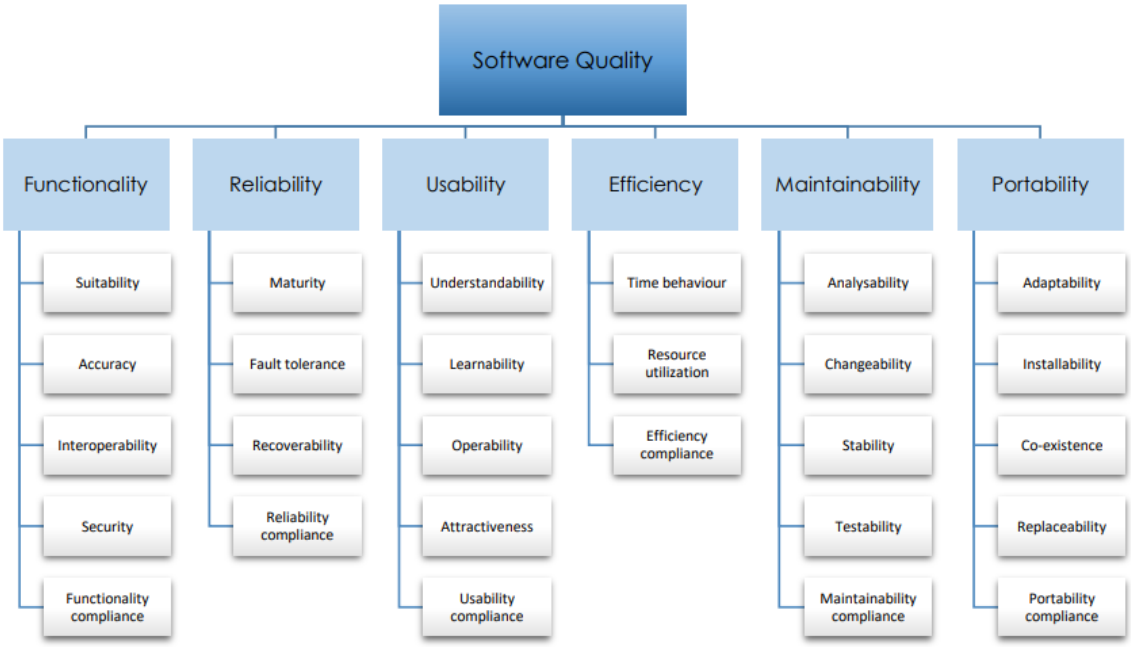
1. Standard, Pratiche, Convenzioni e Metriche

1.1 Definizioni della qualità

Lo standard seguito nel seguente Quality Plan è l'ISO/IEC 9126. Le norme riportate in questo standard descrivono un modello di qualità del software, definiscono le caratteristiche che la determinano e propongono metriche per la misurazione. Le norme relative alla qualità del software emesse da questo standard si dividono in quattro quattro parti:

- Modello della qualità del software;
- Metriche esterne per la qualità;
- Metriche interne per la qualità;
- Metriche per la qualità in uso.

Il modello prevede che per ciascuna di esse siano associate sotto-caratteristiche, dette anche attributi. Nel grafico sottostante saranno mostrate le caratteristiche e gli attributi del software proposti dal modello.



1.1.1 Functionality

La funzionalità rappresenta la capacità del software di fornire funzioni implicite, necessarie per operare in un determinato contesto. Gli attributi del software richiesti da questa caratteristica sono:

- **Suitability**: rappresenta la capacità di un prodotto software di fornire un appropriato insieme di funzioni che permettano agli utenti di svolgere determinati task e di raggiungere gli obiettivi prefissati;
- **Accuracy**: rappresenta la capacità di un prodotto software di fornire i risultati o gli effetti attesi con il livello di precisione richiesta;

1.1.2 Reliability

L'affidabilità rappresenta la capacità di un prodotto software di mantenere il livello di prestazione quando viene utilizzato in condizioni specificate. Possibili limitazioni all'affidabilità del software possono essere causate da errori nei requisiti, nella progettazione e nel codice. Le evidenze di tali errori possono essere rilevate a seconda delle condizioni in cui il prodotto è utilizzato oppure alle opzioni scelte, piuttosto che al momento in cui è utilizzato. L'attributo richiesto in questa caratteristica è il **Fault tolerance**: rappresenta la capacità di un prodotto software di mantenere il livello di prestazioni in caso di errori nel software o di violazione delle interfacce specificate.

1.1.3 Usability

L'usabilità rappresenta la capacità di un prodotto software di essere comprensibile, di poter essere studiato, di risultare attraente da parte di un utente in certe condizioni. Gli attributi di questa caratteristica sono:

- **Understandability**: rappresenta la capacità di un prodotto software di permettere all'utente di capire le sue funzionalità e come poterla utilizzare con successo per svolgere particolari task in determinate condizioni di utilizzo. È influenzato dalla documentazione disponibile e dall'impressione iniziale che si riceve dal prodotto;
- **Operability**: rappresenta la capacità di un prodotto software di permettere all'utente di utilizzarlo e di controllarlo. Gli aspetti relativi alla funzionalità, modificabilità, adattabilità ed installabilità del software possono influire sull'operabilità del prodotto. L'operabilità del software fa riferimento anche alle aspettative dell'utente sulla sua controllabilità, tolleranza ai guasti e conformità;
- **Attractiveness**: rappresenta la capacità di un prodotto software di risultare "attraente" per l'utente. La qualità è relativa alla progettazione dell'aspetto grafico delle sue interfacce, all'utilizzo dei colori e delle immagini;

1.1.4 Efficiency

L'efficienza rappresenta la capacità di un prodotto software di realizzare le funzioni richieste nel minor tempo possibile ed utilizzando nel miglior modo le risorse necessarie, quando opera in determinate condizioni. Le risorse includono altri prodotti software e la configurazione hardware e software del sistema. Gli attributi relativi all'efficienza del software sono:

- **Time behaviour**: rappresenta la capacità di un sistema software di fornire appropriati tempi di risposta, tempi di elaborazione e quantità del lavoro eseguendo le funzionalità previste sotto determinate condizioni di utilizzo;
- **Resource utilization**: rappresenta la capacità di un prodotto software di utilizzare un appropriato numero e tipo di risorse quando esegue le funzionalità previste sotto determinate condizioni di utilizzo. Le persone sono incluse nella definizione di risorse.

1.1.5 Maintainability

La manutenibilità rappresenta la capacità di un prodotto software di essere modificato. Le modifiche possono includere correzioni o adattamenti del software a modifiche negli ambienti, nei requisiti e nelle specifiche funzionali. Gli attributi previsti per la manutenibilità sono:

- **Changeability**: rappresenta la capacità di un prodotto software di consentire lo sviluppo di modifiche al software originale. L'implementazione include modifiche al codice, alla progettazione ed alla documentazione. Nel caso in cui le modifiche debbano essere fatte dagli utenti, la modificabilità può influire sull'operabilità del prodotto;
- **Testability**: rappresenta la capacità di un prodotto software di consentire la verifica e la validazione del software modificato, cioè di eseguire i test;

1.1.6 Portability

La portabilità rappresenta la capacità di un prodotto software di poter essere trasportato da un ambiente ad un altro. L'ambiente include aspetti organizzativi e tecnologici. L'unico attributo previsto è **Adaptability**: rappresenta la capacità di un prodotto software si essere adattato a differenti ambienti senza richiedere azioni specifiche diverse da quelle previste dal software per tali attività. L'adattabilità include la scalabilità delle capacità interne del prodotto (campi delle schermate, tabelle, volumi delle transazioni, formato dei report, ecc.);

1.2 Metriche Interne

Non sono previste delle metriche interne.

1.3 Metriche Esterne

Non sono previste delle metriche esterne.

1.4 Standard per la documentazione

1.4.1 Standard per il processo di documentazione

Per il processo di realizzazione dei documenti di progetto verranno seguiti i seguenti passi:

1. Creare una versione iniziale del documento;
2. Effettuare una revisione della versione;
3. Aggiungere eventuali cambiamenti individuati nella fase di revisione;
4. Scrivere una nuova versione del documento;
5. Ripetere i passi da 2 a 4 fin quando il documento non viene approvato;
6. Realizzare la versione finale del documento;
7. Apportare le correzioni al layout;
8. Sottomettere il documento sulla piattaforma e-learning (se necessario).

1.4.2 Standard per i documenti

1.4.2.1 Insieme di Stili di Base per i Documenti

La stesura dei documenti sarà guidata da template forniti dal PM. In ogni template saranno definiti la struttura e gli stili che i membri del team devono seguire per redigere il documento. Tutta la documentazione dovrà essere scritta in lingua italiana (esclusi diagrammi che richiedono la lingua inglese a causa del software già esistente) utilizzando il software di word processing Microsoft Word. Ogni documento deve contenere:

- Un frontespizio in cui sia presente:
    - Il logo del progetto;
    - Il titolo del documento (acronimo e nome per esteso);
    - La data dell'ultima modifica;

  - La versione del documento;
  - Destinatario;
  - Chi lo ha presentato;
  - Chi l'ha approvato.
- Una Revision history in cui sia presente:
  - Data della stesura del documento;
  - La versione del documento;
  - La descrizione di eventuali cambiamenti introdotti;
  - Il nome dell'autore del documento;
- Un indice dei contenuti.

Ogni pagina deve avere:

- Un' intestazione in cui sia presente:
  - Il logo del dipartimento di informatica dell'Università di Salerno;
  - La seguente intestazione:
    - Laurea Magistrale in informatica- Università di Salerno Corso di Gestione dei Progetti Software – Prof.ssa F. Ferrucci
- Un piè di pagina in cui sia presente:
  - Il titolo del documento sulla sinistra. Il nome del file deve contenere SiglaDocumento_SiglaProgetto_Vers.x.yz;
  - Il numero di pagina sulla destra.

Il font dei documenti dovrà rispettare le seguenti indicazioni:

	Font	Grandezza	Grassetto	Corsivo	Sottolineato	Colore	Allineamento
Titolo Documento	Century Gothic	48	No	No	No	Blu	Destra
Sottotitolo Documento	Garamond	20	No	No	No	Blu	Destra
Titolo Capitoli	Century Gothic	18	No	No	Si	Blu	Sinistra
Titolo Paragrafi	Garamond	13	Si	No	No	Nero	Sinistra
Sottotitoli Paragrafi	Garamond	12	Si	Si	No	Nero	Sinistra

Testo	Garamond	12	No	Se necessario	No	Nero	Giustificato
Intestazione	Garamond	12	No	No	No	Nero	Allinea al centro
Piè di pagina	Century Gothic	8	No	No	No	Blu	Destra
Intestazione Tabelle	Century Gothic	12	Si	No	No	Bianco	Allinea al centro
Contenuto Tabelle	Century Gothic	11	No	No	No	Nero	Allinea al centro
Sommario	Century Gothic	11	No	No	No	Nero	Sinistra

1.4.2.2 Dettagli Colori

Di seguito vengono descritti i dettagli dei colori utilizzati:

- Blu, colore: 1, 50% più scuro;
- Nero, colore: automatico;
- Bianco: colore: sfondo 1.

1.4.2.3 Stile Tabella

Di seguito vengono riportate le caratteristiche degli stili delle tabelle:

- Tabella: griglia 5 scura – colore 1;
- Sfondo intestazione: Blu, colore: 1, 25% più scuro;
- Testo intestazione: Bianco: colore: sfondo 1.

1.4.2.4 Frontespizio

Ogni documento deve presentare un frontespizio standard che presenta le seguenti caratteristiche:

- Il logo del progetto in alto e centrato;
- Il titolo del documento (acronimo e nome per esteso) immediatamente sotto al logo;
- Una tabella riassuntiva in cui siano presenti:
    - La data dell'ultima modifica;
    - La versione del documento;

- Destinatario;
- Chi lo ha presentato;
- Chi l'ha approvato

1.4.2.5 Regole sugli Identificatori dei Documenti

L'assegnamento dei nomi ai documenti prodotti durante le fasi di sviluppo del software è molto importante per la tracciabilità dei documenti stessi; pertanto la sintassi di base da seguire per identificare i documenti è la seguente:

<anno corrente>_<sigla documento>_<numero del gruppo>_<nome progetto>_<cognome PM>

Di seguito è riportato un esempio:

2021_BC_C11_MyBomber_Mauro

È importante, inoltre, mantenere la tracciabilità anche per gli artefatti; per gli artefatti il modello è:

<acronimo artefatto>_<numero artefatto>

Questi sono gli acronimi scelti per i documenti:

- RF: Requisito Funzionale;
- RNF: Requisito non Funzionale;
- SC: Scenario;
- UC: Use Case;
- UCD: Use Case Diagram;
- CD: Class Diagram;
- SD: Sequence Diagram;
- NP: Navigation Path;
- MU: Mock-up.

1.4.2.6 Regole Cambiamenti Versioni di un Documento

Ogni qualvolta viene apportata una modifica sostanziale ad un documento deve essere aggiornata la Revision History ad esso associato. Nella Revision History viene specificato la data della modifica, la versione del documento, la descrizione della modifica e gli autori. In questo modo sarà possibile mantenere una versione distinta del documento per ogni entry della revision history; ogni versione del documento avrà la forma x.y, dove x rappresenta la versione del documento e y la sotto versione che viene a crearsi quando la modifica al documento non è particolarmente ampia. In questo modo, avendo

a disposizione una copia per ogni versione del documento, possiamo sfruttare lo strumento di confronto, messo a disposizione da Microsoft Word, per poter visivamente costatare ciò che differenzia le due versioni.

1.4.3 Standard Scambio Documenti

I documenti devono essere realizzati tramite il programma Microsoft Word, nello specifico la versione 2019, in modo da non avere nessun problema di compatibilità. Tutti i documenti vengono condivisi tra i membri del team e il PM utilizzando One Drive. La sottomissione dei documenti, invece, avviene mediante la piattaforma e-learning del Dipartimento di Informatica.

1.5 Standard per gli Artefatti

1.5.1 Convenzione Requisiti Funzionali

I requisiti funzionali, come anticipato nelle precedenti sezioni, avranno come acronimo RF. La descrizione dei RF deve seguire la seguente specifica:

RF_<numero del requisito>

1.5.2 Convenzione Requisiti Non Funzionali

L'acronimo dei requisiti non funzionali è RNF con numero identificativo progressivo.
L'identificazione dei requisiti non funzionali dovrà fare riferimento al modello FURPS+, ma solamente per le categorie ritenute di interesse per lo sviluppo del progetto.

1.5.3 Convenzione Scenari

Ogni scenario deve far riferimento ad uno e un solo requisito. Il nome dello scenario deve essere costituito da:

SC_ X: <nome dello scenario>, con x numero del requisito a cui fa riferimento.

Le regole per la realizzazione degli scenari sono di seguito riassunte:

- La tabella per ogni scenario deve essere composta da: nome scenario, partecipanti, flusso degli eventi.
- I partecipanti devono essere preceduti dal rispettivo nome proprio.
- Il flusso degli eventi deve iniziare con l'interazione di un partecipante.

- Il flusso degli eventi deve essere strutturato in modo da mostrare le operazioni dei partecipanti allineate a sinistra.
- Il flusso degli eventi deve essere strutturato in modo da mostrare le operazioni del sistema allineate a destra. (se si hanno problemi di resa grafica, il flusso può essere anche descritto in successione, precisando sempre l'attore che effettua l'operazione).
- Nel flusso degli eventi le operazioni dell'utente devono iniziare con il nome proprio di un partecipante.
- Nel flusso degli eventi le operazioni del sistema devono iniziare con "Il sistema…" o con il nome del sistema stesso.

1.5.4 Convenzione Use Case

Ogni use case deve far riferimento ad uno e un solo requisito. Il nome dello use case deve rispettare questo modello:

UC_ X: <nome dello use case>, con X numero del requisito funzionale a cui fa riferimento.

Le regole per la realizzazione degli use case sono di seguito riassunte:

- La tabella per ogni caso d'uso deve essere composta secondo il template allegato;
- Il nome del caso d'uso deve includere un verbo e deve essere univoco;
- Il nome del caso d'uso deve indicare cosa intende fare l'attore;
- Il nome dell'attore deve essere un sostantivo, che indica un ruolo rispetto all'uso del sistema.
- I nomi degli attori e dei casi d'uso devono basarsi su elementi del dominio dell'applicazione, anche i termini del flusso di eventi deve far riferimento al dominio del problema;
- Il flusso degli eventi deve iniziare con l'interazione dell'attore (triggering event);
- Un caso d'uso deve descrivere una transizione utente completa;
- Un caso d'uso non deve descrivere un'interfaccia del sistema, meglio descrivere con mock-up;
- Fare riferimento ai mock-up scrivendo nel punto in cui si fa riferimento all'interfaccia grafica "cfr. (MU_x)" dove x assume i valori 1,2,3…
- La descrizione di un caso d'uso non deve superare le 2 pagine, altrimenti deve essere decomposto in casi d'uso più piccoli;
- Nel caso d'uso base deve essere invocato il caso d'uso incluso in un punto specifico;
- L'evento che determina l'attivazione del caso d'uso che estende deve essere indicato nella condizione di ingresso del caso d'uso che estende;

- Non bisogna usare la forma passiva (le relazioni casuali tra le varie parti del flusso degli eventi devono essere chiari);
- Descrivere tutti i flussi di eventi (non solo quello principale);
- Definire i termini importanti nel glossario.

Di seguito è riportato il template che deve essere usato per gli use case:

Identificativo UC_xx_yy	**Nome del caso d'uso**	Data	gg/mm/aaaa
		Versione	0.00.000
		Autore	Nome Cognome
Descrizione	Descrizione generale del caso d'uso		
Attore Principale	Nome		
Attori secondari	Nome		
Entry Condition	Descrizione condizione di entrata		
Exit Condition On success	Descrizione di uscita in caso di successo		
Exit Condition On failure	Descrizione di uscita in caso di fallimento		
Rilevanza/User Priority	Priorità attribuita al caso d'uso		
Frequenza Stimata	N/giorno		
Extension point	PUNTO DI ESTENSIONE DI ALTRO UC		
Generalization of	Generalizzazione di altro UC		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Attore	Descrizione step 1	
2	Sistema:	Descrizione step 2	
3	Attore	Descrizione step 3	
4	Sistema:	Descrizione step 4	
I Scenario/Flusso di eventi Errore: il sistema non riesce ad inviare la richiesta			
4.a1	Sistema:	Descrizione step flusso di errore	
4.a2	Sistema:	Descrizione step flusso di errore	
I Scenario/Flusso di eventi Alternativo: dati non corretti			
4.b1	Sistema:	Descrizione flusso alternativo	
4.b2	Sistema:	Descrizione flusso alternativo	

1.5.5 Convenzione Use Case Diagram

Gli use case diagram (UCD) devono far riferimento ad un package di requisiti.

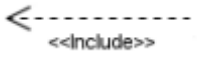
Il nome deve rispettare il formato che segue:

UCD_<nome UCD>

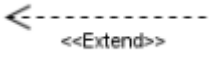
Le regole da rispettare per la stesura degli use case diagram sono le seguenti:

- Il diagramma dei casi d'uso deve essere composto da attori (omini stilizzati) e casi d'uso (ovali). Gli attori devono essere collegati con una linea continua ai casi d'uso a cui partecipano. Il confine del sistema deve essere indicato tracciando un rettangolo (o package) che racchiude tutti i casi d'uso;
- Fornire i diagrammi dei casi d'uso a diversi livelli di astrazione;
- L'attore principale dovrebbe essere posizionato al lato sinistro del rettangolo;
- Sotto l'attore deve essere indicato il nome dell'attore;
- Gli attori secondari devono essere posizionati al lato destro del rettangolo;
- Prima del nome dell'attore che indica un sistema deve essere aggiunto lo stereotipo <>;
- Applicare <> quando si conosce esattamente quando invocare il caso d'uso;
- Applicare <> quando un caso d'uso può essere invocato in varie parti del flusso di eventi del caso d'uso;
- La generalizzazione degli attori deve essere indicata con una freccia che va dall'attore specializzato verso l'attore generico.
- La generalizzazione dei casi d'uso si rappresenta con una freccia che va dal caso d'uso specializzato verso quello generico. Di seguito è riporta l'esempio della freccia usata negli ultimi due punti elencati:

- L'inclusione dei casi d'uso va fatta tramite una freccia con il label <<include>>. La freccia va dal caso d'uso incorporante a quello incluso. Di seguito viene riportato un esempio della freccia:



- L'estensione dei casi d'uso va fatta tramite una freccia con il label <<extend>>. La freccia va dal caso d'uso che estende a quello esteso. Di seguito viene riportato un esempio della freccia:

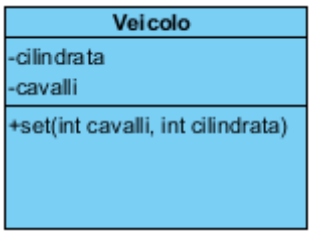


1.5.6 Convenzione Class Diagram

Il nome del class diagram deve avere questa struttura:

CD_<nome del class diagram>

Con questo diagramma si vanno a specificare ad alto livello le classi che faranno parte del sistema e le loro interazioni. Una classe viene rappresentata come segue:

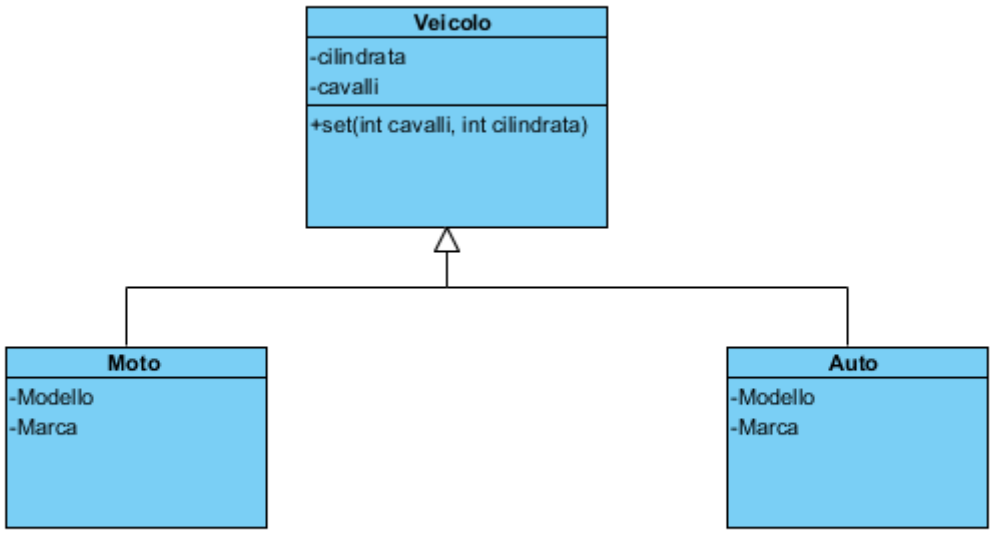


La visibilità di ogni attributo o metodo è specificata in UML attraverso l'uso di vari simboli:

- "-" visibilità privata: l'attributo è accessibile solo dall'interno della classe usando i propri metodi;
- "+" visibilità pubblica: l'attributo o il metodo è accessibile anche dall'esterno;
- "#" visibilità protetta: l'attributo o il metodo viene ereditato da tutte le classi da questa derivate.

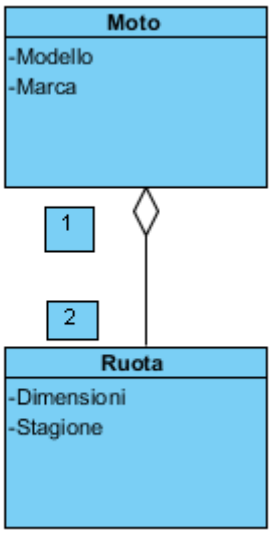
1.5.6.1 Relazioni tra Classi ed Oggetti

- Generalizzazione-specializzazione: con questa relazione è possibile creare una superclasse da cui far derivare in seguito classi figlie. Evitando di specializzare da subito tutti gli attributi di una classe, si ha il vantaggio di poter inquadrare sin dall'inizio il domino di applicazione. Questa relazione si traduce nei linguaggi di programmazione orientati agli oggetti con l'uso dell'ereditarietà.



- Aggregazione: con questa relazione si possono creare aggregati di oggetti; il vantaggio di usare questa relazione consiste nel fatto che, in presenza di un oggetto composto da più parti, non andiamo a gestire l'oggetto nella sua totalità ma in funzione delle sue parti; in questo modo si va a gestire piccole classi piuttosto che una sola grande classe.

Un'ulteriore aggiunta a questa relazione è data dalla molteplicità. La molteplicità, posta in prossimità della freccia che rappresenta l'aggregazione, indica il numero di occorrenze necessitiamo di una classe per ogni aggregazione; se si vogliono indicare molte occorrenze si utilizza il simbolo *.

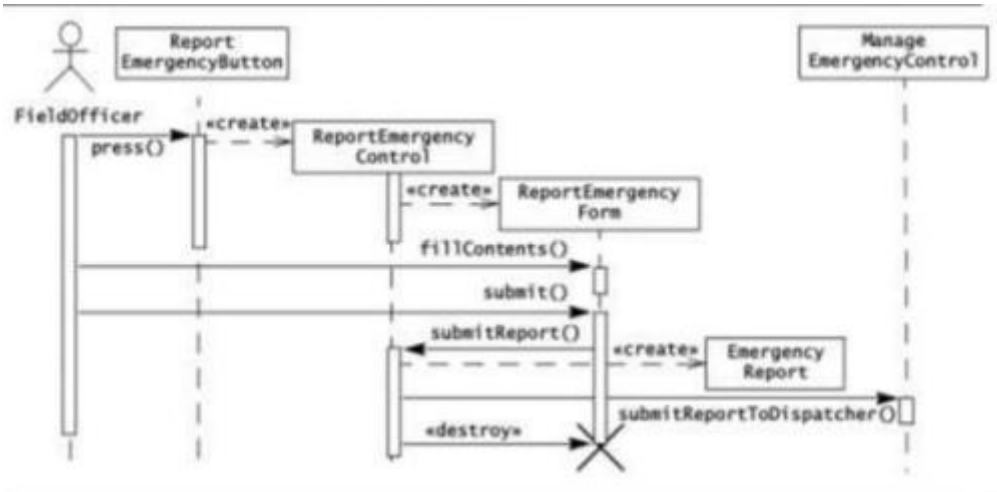


- Associazione: con questa associazione si esprime i legami che ci sono fra le classi. È caratterizzata da:
    - un nome, rappresenta il nome che attribuiamo all'associazione;
    - una molteplicità, ha lo stesso significato dell'aggregazione; o una direzione, indica il verso di lettura dell'associazione, se abbiamo una linea senza frecce indica la bidirezionalità;

1.5.7 Convenzione Sequence Diagram

Ogni sequence diagram deve far riferimento ad uno e un solo requisito. Il nome del sequence diagram deve rispettare questo modello:

SQ_X: <nome del sequence> , con X numero del requisito funzionale a cui fa riferimento.



1.5.7.1 Layout

- Le colonne rappresentano gli oggetti che partecipano al caso d'uso:
    - La prima colonna deve corrispondere all'attore che ha avviato il caso d'uso;
    - La seconda colonna deve corrispondere ad un oggetto boundary con cui l'attore interagisce per iniziare il caso d'uso;
    - La terza colonna dovrebbe corrispondere all'oggetto control che gestiste il caso d'uso.

1.5.7.2 Creazione

- Gli oggetti control vengono creati all'inizio del sequence diagram e si estendono per tutta la durata dello stesso;

- Gli oggetti control possono creare altri oggetti boundary e possono interagire con altri oggetti control;
- In cima al diagramma (vedi figura 5-8, a destra dell'attore) si trovano gli oggetti che esistono prima che il flusso abbia inizio (nell'esempio l'attore, FieldOfficer, clicca sul bottone ReportEmergencyButton; il bottone esiste dapprima che l'attore invii il primo messaggio).

1.5.7.3 Accesso

- Gli oggetti Entity sono accessibili da oggetti Control e Boundary;
- Gli oggetti Entity non possono chiamare oggetti Control o oggetti Boundary.

1.5.7.4 Istanze delle classi

- Le istanze vengono rappresentate da rettangoli con il nome della classe e l'identificatore dell'oggetto sottolineato, oppure, semplicemente con un nome dal quale si evince che si sta considerando un'istanza della classe (es. ReportEmergencyButton).

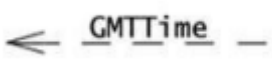
1.5.7.5 Attori

- Il sequence diagram deve essere composto da attori (es. l'omino stilizzato dell'esempio 5-8). È necessario indicare il nome dell'attore;
- L'attore tempo deve essere denominato Time;
- Gli attori vengono posizionati sulla sinistra, con le frecce di interazione verso gli oggetti del sistema;
- Gli attori possono anche non essere riportati nel caso in cui il caso d'uso venga avviato da un altro caso d'uso.

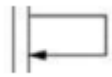
1.5.7.6 Messaggi

- Le frecce orizzontali tra le colonne rappresentano i messaggi inviati da un oggetto a un altro. I messaggi sono rappresentati come frecce da un attore a un oggetto, o tra due oggetti:

- I messaggi di return sono rappresentati con frecce tratteggiate:



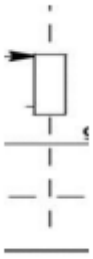
- Una freccia circolare (che parte e arriva nella stessa box di attivazione) rappresenta un messaggio che ha per destinatario lo stesso mittente. Rappresentato nel seguente modo:



- Gli oggetti creati durante l'interazione sono preceduti da un messaggio di <>;
- Gli oggetti distrutti durante l'interazione sono evidenziati con una croce e preceduti da un messaggio <>.

1.5.7.7 Lifelines

Le lifelines sono linee tratteggiate verticali che partono dal rettangolo rappresentativo dell'oggetto e giungono fino in fondo al diagramma. Indicano il periodo temporale di vita dell'oggetto, della creazione alla distruzione.



1.5.8 Convenzione Statechart Diagram

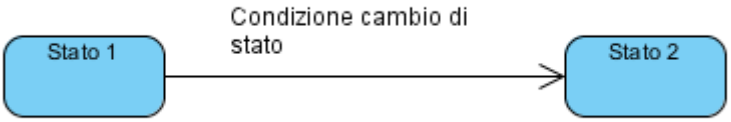
Il nome dello Statechart Diagram deve rispettare questo modello:

SCD_X <nome entità coinvolta>, Con x numero identificativo.

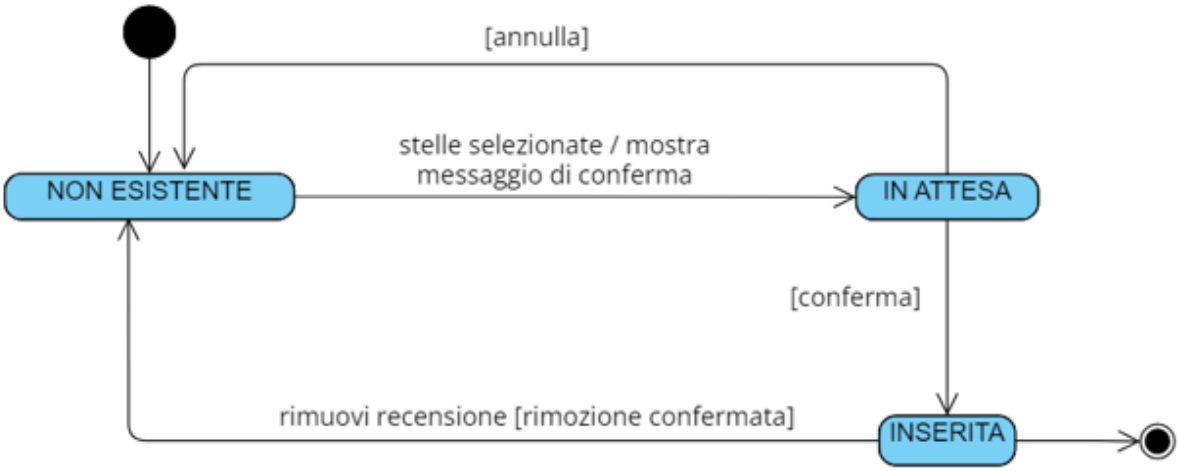
Con questi diagrammi si descrive un comportamento dinamico del sistema, modellando dinamicamente ciò che accade in un determinato oggetto. In sostanza si rappresenta il comportamento di un determinato oggetto di una classe in relazione anche ad input esterni. Questi diagrammi sono caratterizzati da quattro componenti principali:

- **Stato iniziale**: è lo stato iniziale dell'oggetto; è rappresentato con un cerchio colorato di nero;
- **Stato generico**: rappresenta un generico stato assunto dall'oggetto. È rappresentato con un rettangolo con gli angoli stondati;

- **Transazioni**: collegano uno stato sorgente ad uno stato destinazione nel quale l'oggetto può andare. Sono rappresentate con delle frecce con all'estremità una freccia che punta verso lo stato destinazione;
- **Stato finale**: è lo stato finale dell'oggetto; è rappresentato dal simbolo dello stato iniziale inscritto in un cerchio più grande a sfondo bianco.



Di seguito è riportato un esempio di statechart diagram:



1.5.9 Convenzione Mockup

Ogni mock-up deve essere specifico a uno scenario per poter facilitare la lettura del flusso degli eventi dello scenario di riferimento. Il nome del mock-up deve rispettare questo modello:

UI_ X: <nome del mockup>, con X numero del requisito funzionale a cui fa riferimento.

1.6 Standard Codifica

Per la stesura del codice si dovranno rispettare le regole e le convenzioni definite dai team member nell'Object Design Document. Queste regole verranno inserite nel tool di controllo della qualità CheckStyle per verificare successivamente la qualità del codice.

1.7 Standard e Pratiche per il Testing

Per il testing si andranno a rispettare la pianificazione delle attività dedicate al testing, seguendo e adottando le regole illustrate ai team member durante le lezioni di Ingegneria del Software e presenti sul libro di riferimento del corso.

1.8 Metriche per la valutazione del progetto

Ogni documento che sarà rilasciato sarà sottoposto ad una doppia fase di verifica e validazione. Innanzitutto, sono i responsabili del documento che controlleranno la qualità del documento stesso e ne forniranno una copia al PM; questa prima fase sarà guidata dalle linee guida e checklist che il PM fornirà ai responsabili del documento. Una volta consegnato il documento al PM, quest'ultimo darà via ad una nuova fase di verifica durante la quale controllerà che le checklist siano realmente verificate. Un documento supererà questo controllo di qualità se almeno il 75% dei parametri contenuti nelle checklist saranno soddisfatti.

2. Revisioni del Software

Artefatto	Data di Revisione	Metodo di Revisione	Azioni ulteriori
RAD Funzionale	25/11/2021	Lettura	Feedback ed eventuali segnalazioni di modifiche
RAD Completo	03/12/2021	Lettura	Feedback ed eventuali segnalazioni di modifiche
SDD	09/12/2021	Lettura	Feedback ed eventuali segnalazioni di modifiche
TCS	13/12/2021	Lettura	Feedback ed eventuali segnalazioni di modifiche
ODD	24/12/2021	Lettura	Feedback ed eventuali segnalazioni di modifiche
Implementazione	08/01/2022	Prova del sistema	Feedback ed eventuali segnalazioni di modifiche
TER	20/01/2022	Lettura	Feedback ed eventuali segnalazioni di modifiche

TSR	20/01/2022	Lettura	Feedback ed eventuali segnalazioni di modifiche
TIR	20/01/2022	Lettura	Feedback ed eventuali segnalazioni di modifiche
MI	22/01/2022	Lettura	Feedback ed eventuali segnalazioni di modifiche
MU	22/01/2022	Lettura	Feedback ed eventuali segnalazioni di modifiche

3. Test

Le attività di testing del sistema realizzato vengono eseguite secondo i piani e i criteri definiti all'interno del documento 2021_TP_C11_MyBomber_Mauro.

4. Rapporto sui problemi e azioni correttive

Ogni problematica riscontrata all'interno del gruppo verrà risolta durante i meeting settimanali. È possibile segnalare eventuali criticità direttamente al PM. Una volta individuata una criticità, il PM cercherà una risoluzione al problema in modo da non causare ritardi nelle consegne del progetto e non rovinare il clima all'interno del gruppo.

5. Strumenti, tecniche e metodologie

Il processo di controllo della qualità avviene ogni volta che un team member consegna il proprio task su Trello. Il PM provvederà a revisionare il task e lasciare il proprio feedback. Inoltre, ogni due settimane i team members riceveranno una valutazione sulla qualità degli artefatti prodotti e sulla produttività, in modo da permettergli di migliorarsi nelle settimane successive di lavoro.

6. Controllo dei dati multimediali

Tutti i dati multimediali utili allo sviluppo del progetto, siano essi di supporto per il team o necessari per il prodotto finale, verranno mantenuti in servizi di cloud Storage e Backup (es. One Drive) in modo tale che essi possano essere facilmente reperibili e modificabili da tutti i membri del team, anche se si trovino in mobilità. Il Project Manager potrà anch'esso accedere a tali dati in modo tale da poterli visionare, modificare ed in generale monitorare.

7. Collezione, manutenzione e conservazione dei dati

Tutti i dati del progetto, documenti e software saranno salvati su Trello, sulla cartella condivisa di One Drive, sulla piattaforma e-learning e sui propri PC. Inoltre, su slack saranno condivise le agende, mentre le minute saranno condivise su trello.

8. Training

Sarà svolto un training su Selenium, Maven, GitHub, JUnit, Mockito e bootstrap.

9. Gestione dei Rischi

ID	Nome del Rishio	Descrizione	Piano di Contingenza
R_01	Membri del team non seguono le linee guida fornite	Sono state fornite delle linee guida da seguire per la realizzazione corretta della documentazione	Incontro con i team member per comprendere se il problema sono le linee guida poco chiare e spiegare in modo chiaro cosa c'è che non va nei documenti.
R_02	Qualità del documento	La qualità del documento non è quella che ci si aspettava	Controlli rapidi e precisi sui documenti per segnalare in maniera tempestiva le cose che vanno modificate/migliorate.

10. Glossario

IEC: è un'organizzazione internazionale per la definizione di standard in materia di elettricità, elettronica e tecnologie correlate.

ISO: è la più importante organizzazione a livello mondiale per la definizione di norme tecniche.

Mockup: insieme di immagini che servono a descrivere a grandi linee l'interfaccia grafica del sistema.

Slack: software utilizzato per inviare messaggi in modo istantaneo ai membri del team.

SOW: acronimo di "Statement Of Work" è documento che contiene le specifiche preliminari del progetto. Checklist: elenco esaustivo di cose da verificare nel momento in cui viene prodotto un artefatto.

11. Procedure di aggiornamento del piano

Data	Versione	Descrizione	Autori
28/12/2021	0.1	Prima stesura	Gaetano Mauro
14/01/2022	1.0	Revisione	Gaetano Mauro