

# Software Maintenance - CMMTickets



**UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO**

Caprinali Michele 1087210  
Mazzoleni Gabriel 1086530  
Mazzoleni Raphael 1086531

Indice:

1. [Introduzione](#)
2. [Categorie di manutenzione](#)
3. [Refactoring](#)

## **1- Introduzione**

Nel seguente documento verranno presentate le attività di manutenzione e di refactoring che sono state adottate nello sviluppo del sistema, che sono risultate implicite e integrate durante tutta la durata dello sviluppo

## **2- Categorie di Manutenzione**

Le attività di manutenzione eseguite hanno riguardato maggiormente la manutenzione correttiva e la manutenzione perfettiva. Abbiamo dovuto correggere funzionalità implementate precedentemente in quanto non rispecchiavano quanto dichiarato nella specifica dei requisiti o presentavano degli errori/bug. Sono state assenti attività di manutenzione adattiva.

### 3- Refactoring

L'attività di manutenzione che ha occupato più tempo è stata quella di refactoring, un processo di miglioramento del codice sorgente di un software senza modificarne il comportamento funzionale esterno. Consiste nell'ottimizzare e riorganizzare il codice per renderlo più chiaro, efficiente, manutenibile e flessibile. Le principali tecniche di refactoring utilizzate sono:

- Rinomina di variabili e metodi
- Rimozione di codice duplicato e inutilizzato
- Semplificazione di espressioni condizionali

Di seguito verrà mostrato un esempio di refactoring all'interno del nostro software che aveva una cognitive complexity di 16 e l'abbiamo risolta scomponendo la funzione in sottofunzioni:

```
public void fetchAndDisplayEvents() throws ParseException {
    eventi = EventsDatabase.getAllEvents();
    luoghi = LuoghiDatabase.getAllLuoghi();
    String[] columnNames = {"Nome Evento", "Data", "Luogo", "Città", "Indirizzo"};
    Object[][] data = new Object[eventi.size()][5];

    +1
    1 for (int i = 0; i < eventi.size(); i++) {
        Evento e = eventi.get(i);
        Luogo l;

        +2 (incl 1 for nesting)
        2 for (int y = 0; y < luoghi.size(); y++) {
            data[i][0] = e.getNome();
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
            data[i][1] = sdf.format(e.getData());

            l = luoghi.get(y);

            +3 (incl 2 for nesting)
            3 if (e.getIdLuogo() == l.getIdLuogo()) {
                data[i][2] = l.getNome();
                data[i][3] = l.getIndirizzo();
                data[i][4] = l.getCittà();
            }
        }
    }
}
```

```
public void fetchAndDisplayEvents() throws ParseException {  
    eventi = EventsDatabase.getAllEvents();  
    luoghi = LuoghiDatabase.getAllLuoghi();  
    String[] columnNames = {"Nome Evento", "Data", "Luogo", "Città", "Indirizzo"};  
    Object[][] data = populateEventData();  
  
    tableModel = new DefaultTableModel(data, columnNames);  
    eventTable = createEventTable();  
  
    JScrollPane scrollPane = new JScrollPane(eventTable);  
    scrollPane.setBorder(BorderFactory.createTitledBorder("Lista Eventi"));  
    add(scrollPane, BorderLayout.CENTER);  
}
```

[https://github.com/RaphacePrime/CMM\\_Tickets/commit/11285a2dd22fbb9b9ccbc981032cafd100a34ce5](https://github.com/RaphacePrime/CMM_Tickets/commit/11285a2dd22fbb9b9ccbc981032cafd100a34ce5)