In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

In [3]:

```
# Load dataset
file_path = r"C:\Users\Godfather Haiku\Desktop\quality.csv"
data = pd.read_csv(file_path)
```

In [5]:

```
# Data Cleaning
data['defect_date'] = pd.to_datetime(data['defect_date'], format='%m/%d/%Y', err
data['repair_cost'] = pd.to_numeric(data['repair_cost'], errors='coerce')

# Check for missing values
print("Missing values:\n", data.isnull().sum())
```

```
Missing values:
 defect_id             0
product_id            0
defect_type           0
defect_date         402
defect_location       0
severity              0
inspection_method     0
repair_cost           0
dtype: int64
```

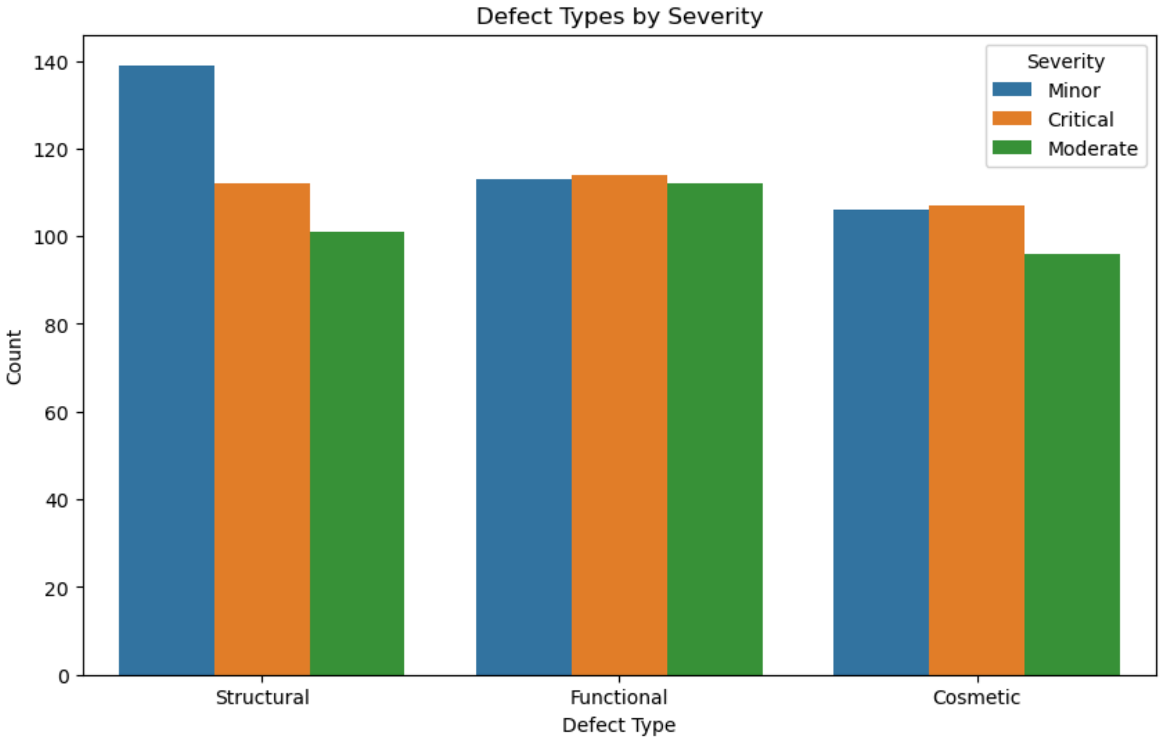
In [6]:

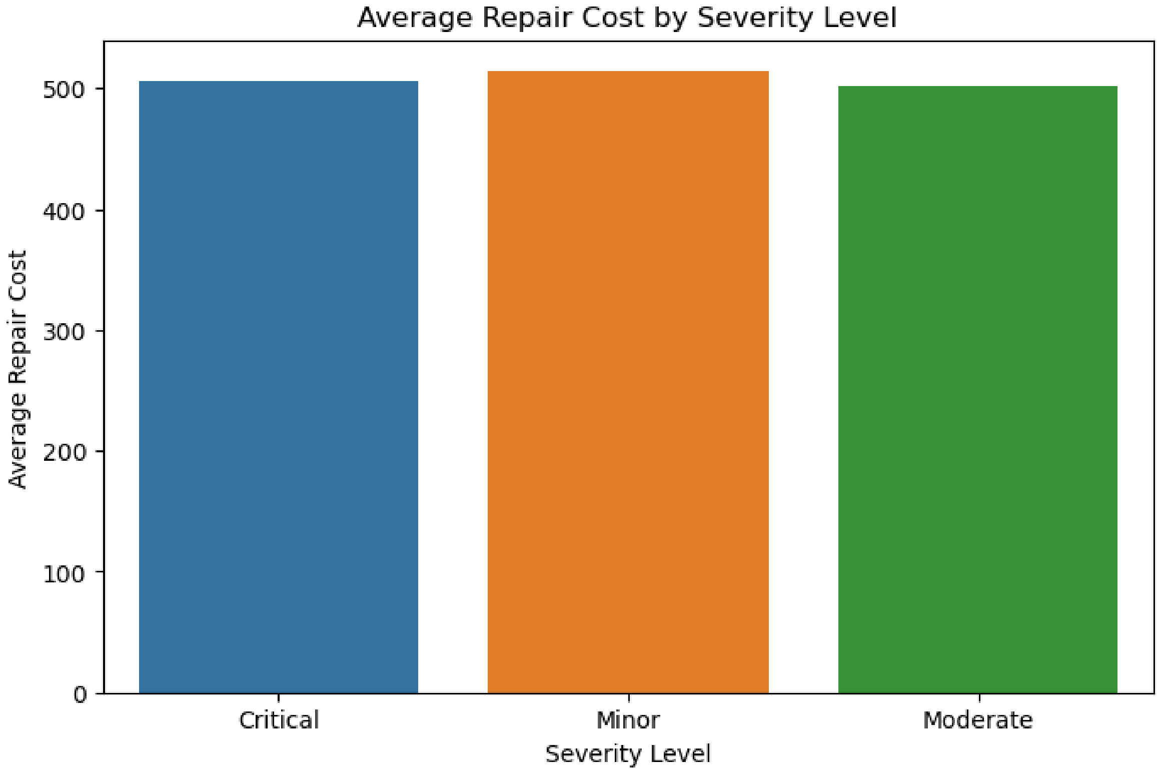
```
# Descriptive Statistics
print("Descriptive statistics:\n", data.describe())


# Defect Frequency Analysis
defect_counts = data['defect_type'].value_counts()
print("Defect Counts:\n", defect_counts)

# Severity Impact on Repair Costs
severity_costs = data.groupby('severity')['repair_cost'].mean()
print("Average Repair Cost by Severity:\n", severity_costs)
```

```
Descriptive statistics:
          defect_id   product_id                    defect_date  repair_cost
count  1000.000000  1000.000000                            598  1000.000000
mean    500.500000    50.837000  2024-04-03 15:48:45.752508416   507.627150
min       1.000000     1.000000            2024-01-13 00:00:00    10.220000
25%     250.750000    26.000000            2024-02-18 00:00:00   270.902500
50%     500.500000    51.000000            2024-03-29 00:00:00   506.430000
75%     750.250000    77.000000            2024-05-19 18:00:00   759.065000
max    1000.000000   100.000000            2024-06-30 00:00:00   999.640000
std     288.819436    29.480935                            NaN   289.623615
Defect Counts:
 defect_type
Structural    352
Functional    339
Cosmetic      309
Name: count, dtype: int64
Average Repair Cost by Severity:
 severity
Critical    505.871622
Minor       514.432877
Moderate    501.634078
Name: repair_cost, dtype: float64
```

In [7]:

```python
# Visualization of Defect Types
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='defect_type', hue='severity')
plt.title('Defect Types by Severity')
plt.xlabel('Defect Type')
plt.ylabel('Count')
plt.legend(title='Severity')
plt.show()
# Visualization of Average Repair Cost by Severity
plt.figure(figsize=(8, 5))
sns.barplot(x=severity_costs.index, y=severity_costs.values)
plt.title('Average Repair Cost by Severity Level')
plt.xlabel('Severity Level')
plt.ylabel('Average Repair Cost')
plt.show()
```





In [8]:

```
# Feature Engineering: Convert categorical variables to dummy variables
data = pd.get_dummies(data, columns=['defect_type', 'severity', 'inspection_meth

# Define features and target variable for predictive modeling
X = data.drop(columns=['repair_cost', 'defect_id', 'product_id', 'defect_date',
y = data['repair_cost']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [9]:

```
# Model Training: Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
```

```
# Predictions
predictions = model.predict(X_test)
```

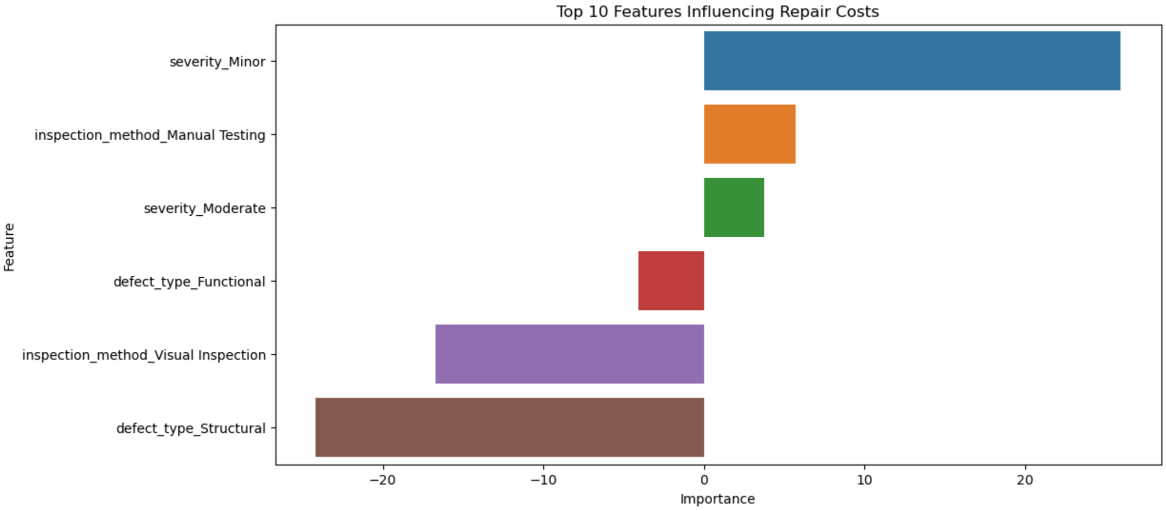
In [10]:

```
# Evaluation: Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, predictions)
print(f'Mean Absolute Error: {mae}')

# Feature Importance Analysis (if needed)
importance = model.coef_
features = X.columns

# Create a DataFrame for feature importance visualization
importance_df = pd.DataFrame({'Feature': features, 'Importance': importance})
importance_df = importance_df.sort_values(by='Importance', ascending=False)
```

```
Mean Absolute Error: 255.07733064142354
```

In [11]:

```
# Visualization of Feature Importance
plt.figure(figsize=(12, 6))
sns.barplot(data=importance_df.head(10), x='Importance', y='Feature')
plt.title('Top 10 Features Influencing Repair Costs')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()
```


Top 10 Features Influencing Repair Costs
severity_Minor
inspection_method_Manual Testing
severity_Moderate
defect_type_Functional
inspection_method_Visual Inspection
defect_type_Structural
Feature
Importance