In [32]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import pickle
```

In [9]:
```
file_path = 'INTELLICATHDATA.csv'
data = pd.read_csv(file_path)

data = data.sample(frac=1, random_state=42).reset_index(drop=True)
```

In [29]:
```
X = data[['catheter_bag_volume', 'flow_rate']]
y = data['time']

# Calculate the correlation coefficients between the features and the target variable
correlation_flow_rate = X['flow_rate'].corr(y)
correlation_bag_volume = X['catheter_bag_volume'].corr(y)
print(f'Correlation between Flow Rate and Time Before Full: {correlation_flow_rate:.4f}')
print(f'Correlation between Catheter Bag Volume and Time Before Full: {correlation_bag_volume:.4f}')
```

```
Correlation between Flow Rate and Time Before Full: -0.6502
Correlation between Catheter Bag Volume and Time Before Full: -0.6644
```

In [10]:
```
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```

In [11]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[11]:
```
LinearRegression
LinearRegression()
```

In [12]:
```
y_val_pred = model.predict(X_val)

mse_val = mean_squared_error(y_val, y_val_pred)
r2_val = r2_score(y_val, y_val_pred)
mae_val = mean_absolute_error(y_val, y_val_pred)
```

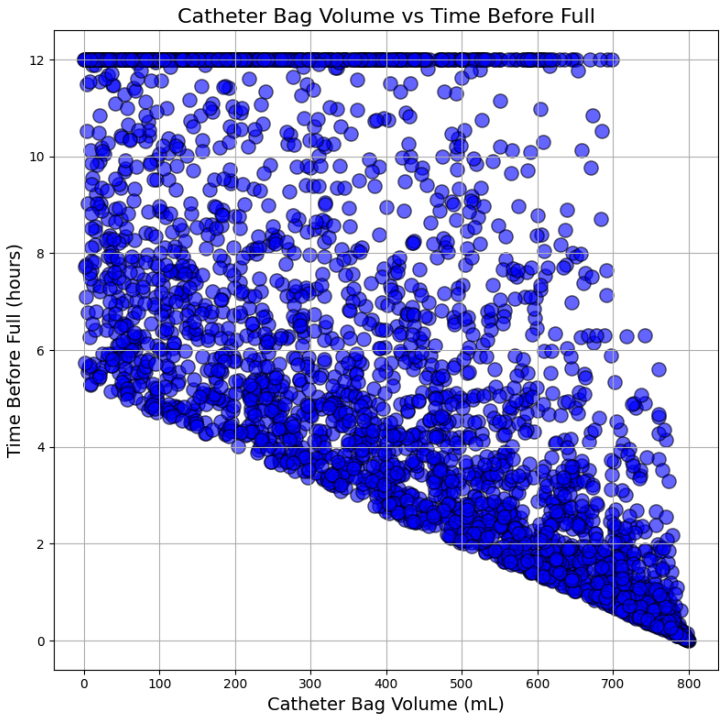
In [13]:
```
y_test_pred = model.predict(X_test)

mse_test = mean_squared_error(y_test, y_test_pred)
r2_test = r2_score(y_test, y_test_pred)
mae_test = mean_absolute_error(y_test, y_test_pred)
```

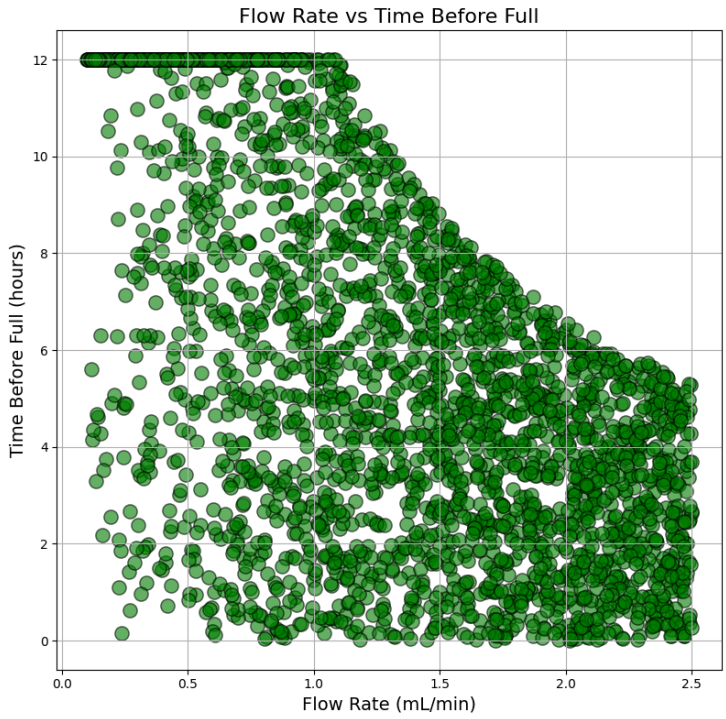
In [26]:
```
print(f'Validation MSE: {mse_val:.4f}')
print(f'Validation R²: {r2_val:.4f}')
print(f'Validation MAE: {mae_val:.4f}')

print(f'Test MSE: {mse_test:.4f}')
print(f'Test R²: {r2_test:.4f}')
print(f'Test MAE: {mae_test:.4f}')  # MAE for test set
```

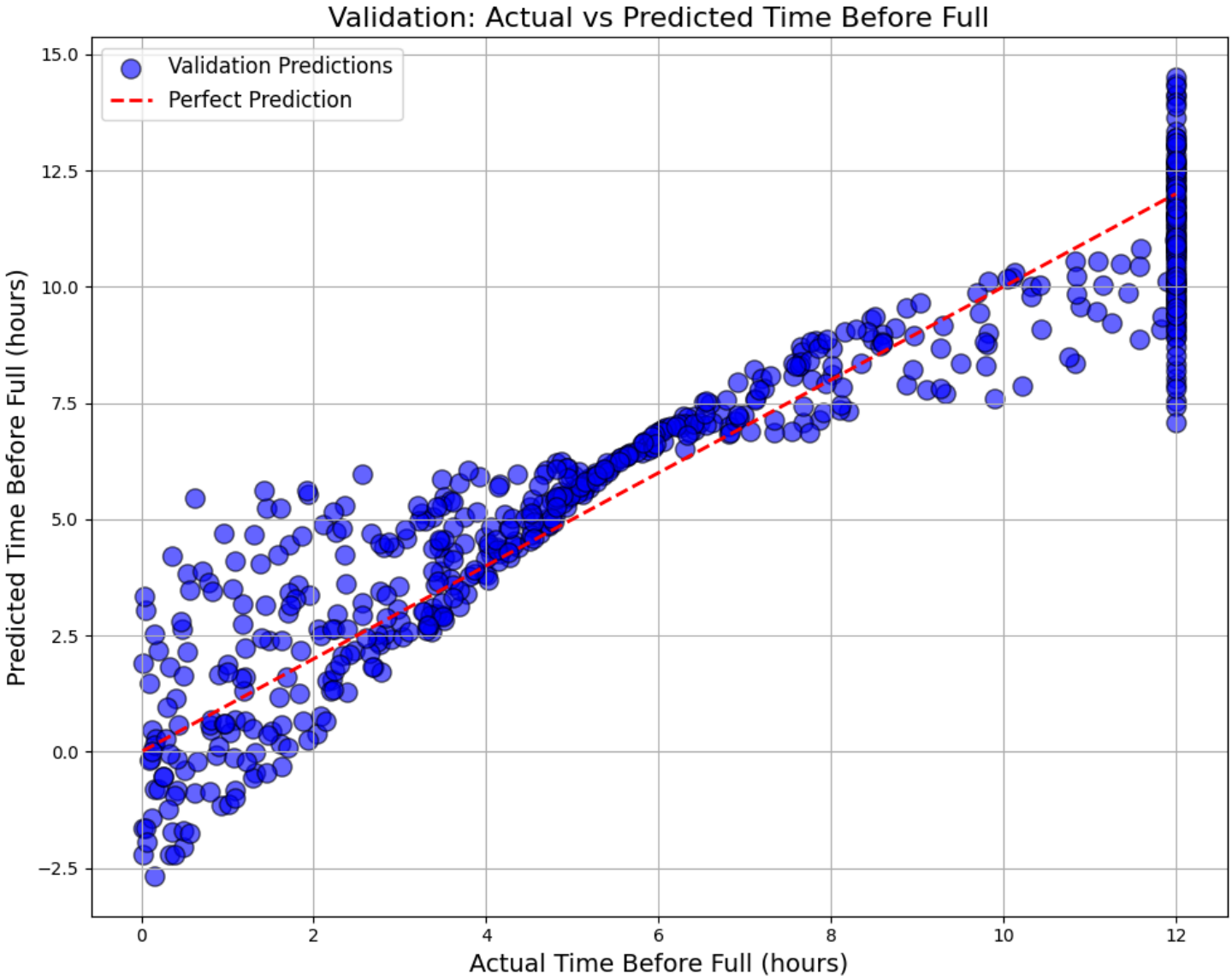
```
Validation MSE: 2.0021
Validation R²: 0.8752
Validation MAE: 1.0844
Test MSE: 2.0463
Test R²: 0.8707
Test MAE: 1.0996
```

In [25]:
```
# Plot the relationships between features and target
plt.figure(figsize=(16, 8))  # Increase the figure size

# Plot 1: Catheter Bag Volume vs Time Before Full
plt.subplot(1, 2, 1)
plt.scatter(X['catheter_bag_volume'], y, alpha=0.6, color='blue', edgecolors='k', s=120)
plt.title('Catheter Bag Volume vs Time Before Full', fontsize=16)
plt.xlabel('Catheter Bag Volume (mL)', fontsize=14)
plt.ylabel('Time Before Full (hours)', fontsize=14)
plt.grid(True)  # Add grid for better readability
plt.tight_layout()
```

```
# Plot 2: Flow Rate vs Time Before Full (Adjusted to match Plot 1)
plt.subplot(1, 2, 2)
plt.scatter(X['flow_rate'], y, alpha=0.6, color='green', edgecolors='k', s=120)
plt.title('Flow Rate vs Time Before Full', fontsize=16)
plt.xlabel('Flow Rate (mL/min)', fontsize=14)
plt.ylabel('Time Before Full (hours)', fontsize=14)
plt.grid(True)  # Add grid for better readability
plt.tight_layout()

# Show the plots
plt.show()

# Visualize the model's predictions on the validation set with better aesthetics
plt.figure(figsize=(10, 8))  # Larger figure size for this plot
plt.scatter(y_val, y_val_pred, alpha=0.6, color='blue', label='Validation Predictions', s=120, edgecolors='k')
plt.plot([min(y_val), max(y_val)], [min(y_val), max(y_val)], color='red', linestyle='--', label='Perfect Prediction', linew
plt.title('Validation: Actual vs Predicted Time Before Full', fontsize=16)
plt.xlabel('Actual Time Before Full (hours)', fontsize=14)
plt.ylabel('Predicted Time Before Full (hours)', fontsize=14)
plt.legend(fontsize=12)
plt.grid(True)  # Add grid for better readability
plt.tight_layout()

# Show the plot
plt.show()
```







In [34]:
```
with open("linear_regression_model4.pkl", "wb") as file:
 pickle.dump(model, file)
```

In [ ]: