Credit Risk Default Prediction - Exploratory Data Analysis

This notebook performs exploratory data analysis on the German Credit Dataset to understand credit default patterns.

In [40]:
```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Set modern style for visualizations
sns.set_theme(style="whitegrid")
%matplotlib inline
```

1. Data Loading and Initial Exploration

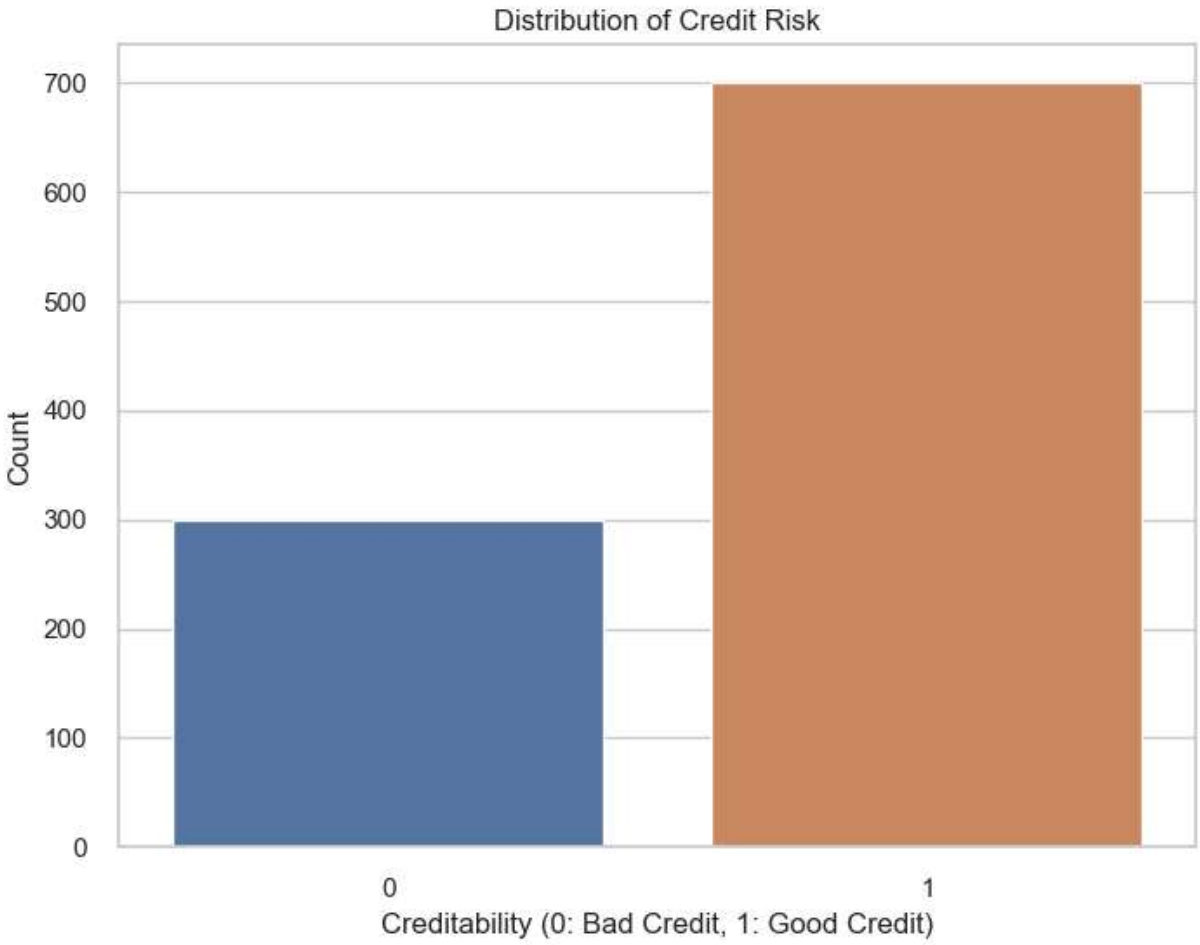
In [41]:
```
# Read the CSV file with correct delimiter
df = pd.read_csv('german.csv', sep=';')

# Display basic information about the dataset
print("Dataset Shape:", df.shape)
print("\nColumn Names:", list(df.columns))
print("\nDataset Info:")
df.info()
```

```
Dataset Shape: (1000, 21)

Column Names: ['Creditability', 'Account_Balance', 'Duration_of_Credit_monthly', 'Pa
yment_Status_of_Previous_Credit', 'Purpose', 'Credit_Amount', 'Value_Savings_Stock
s', 'Length_of_current_employment', 'Instalment_per_cent', 'Sex_Marital_Status', 'Gu
arantors', 'Duration_in_Current_address', 'Most_valuable_available_asset', 'Age_year
s', 'Concurrent_Credits', 'Type_of_apartment', 'No_of_Credits_at_this_Bank', 'Occupa
tion', 'No_of_dependents', 'Telephone', 'Foreign_Worker']

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   Creditability                        1000 non-null   int64
 1   Account_Balance                      1000 non-null   int64
 2   Duration_of_Credit_monthly           1000 non-null   int64
 3   Payment_Status_of_Previous_Credit    1000 non-null   int64
 4   Purpose                              1000 non-null   int64
 5   Credit_Amount                        1000 non-null   int64
 6   Value_Savings_Stocks                 1000 non-null   int64
 7   Length_of_current_employment         1000 non-null   int64
 8   Instalment_per_cent                  1000 non-null   int64
 9   Sex_Marital_Status                   1000 non-null   int64
 10  Guarantors                           1000 non-null   int64
 11  Duration_in_Current_address          1000 non-null   int64
 12  Most_valuable_available_asset        1000 non-null   int64
 13  Age_years                            1000 non-null   int64
 14  Concurrent_Credits                   1000 non-null   int64
 15  Type_of_apartment                    1000 non-null   int64
 16  No_of_Credits_at_this_Bank           1000 non-null   int64
 17  Occupation                           1000 non-null   int64
 18  No_of_dependents                     1000 non-null   int64
 19  Telephone                            1000 non-null   int64
 20  Foreign_Worker                       1000 non-null   int64
dtypes: int64(21)
memory usage: 164.2 KB
```

2. Class Balance Analysis

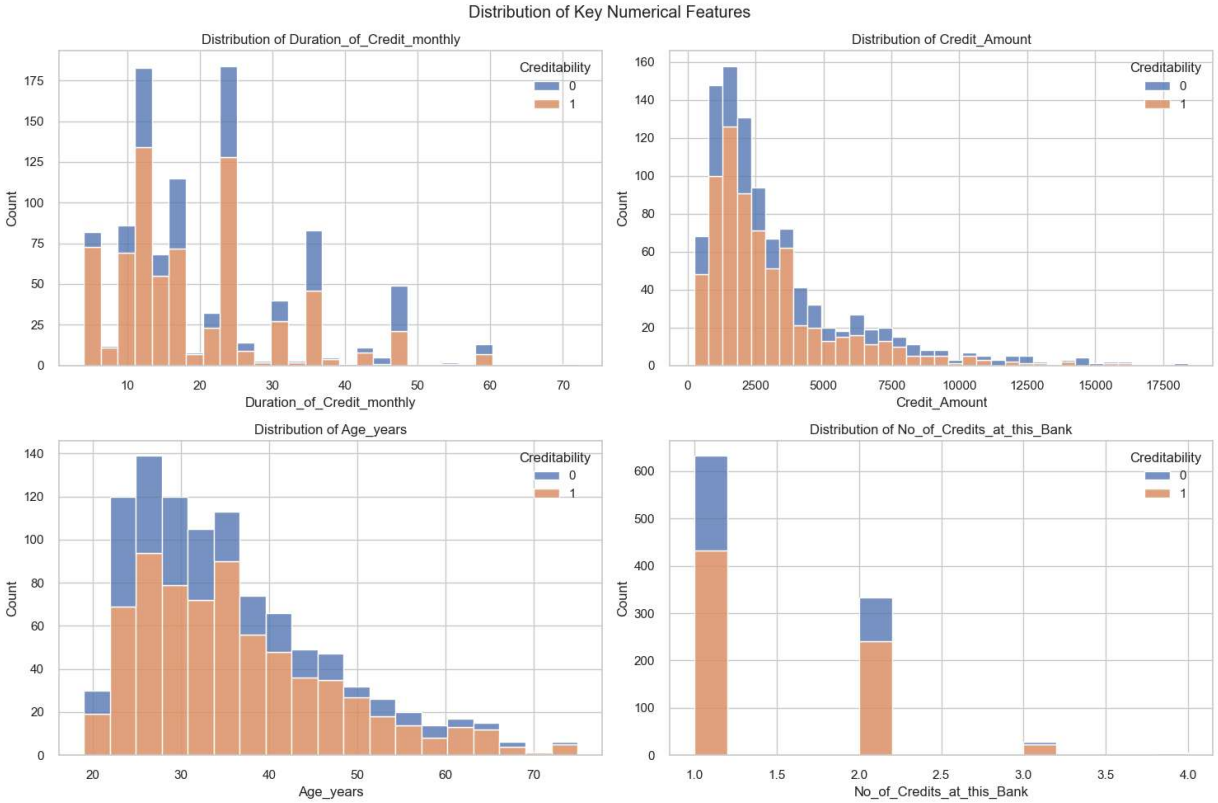
In [42]:
```python
# Check class distribution
class_distribution = df['Creditability'].value_counts()
print("\nClass Distribution:")
print(class_distribution)

# Visualize class distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Creditability')
plt.title('Distribution of Credit Risk')
plt.xlabel('Creditability (0: Bad Credit, 1: Good Credit)')
plt.ylabel('Count')
plt.show()
```

```
Class Distribution:
Creditability
1    700
0    300
Name: count, dtype: int64
```



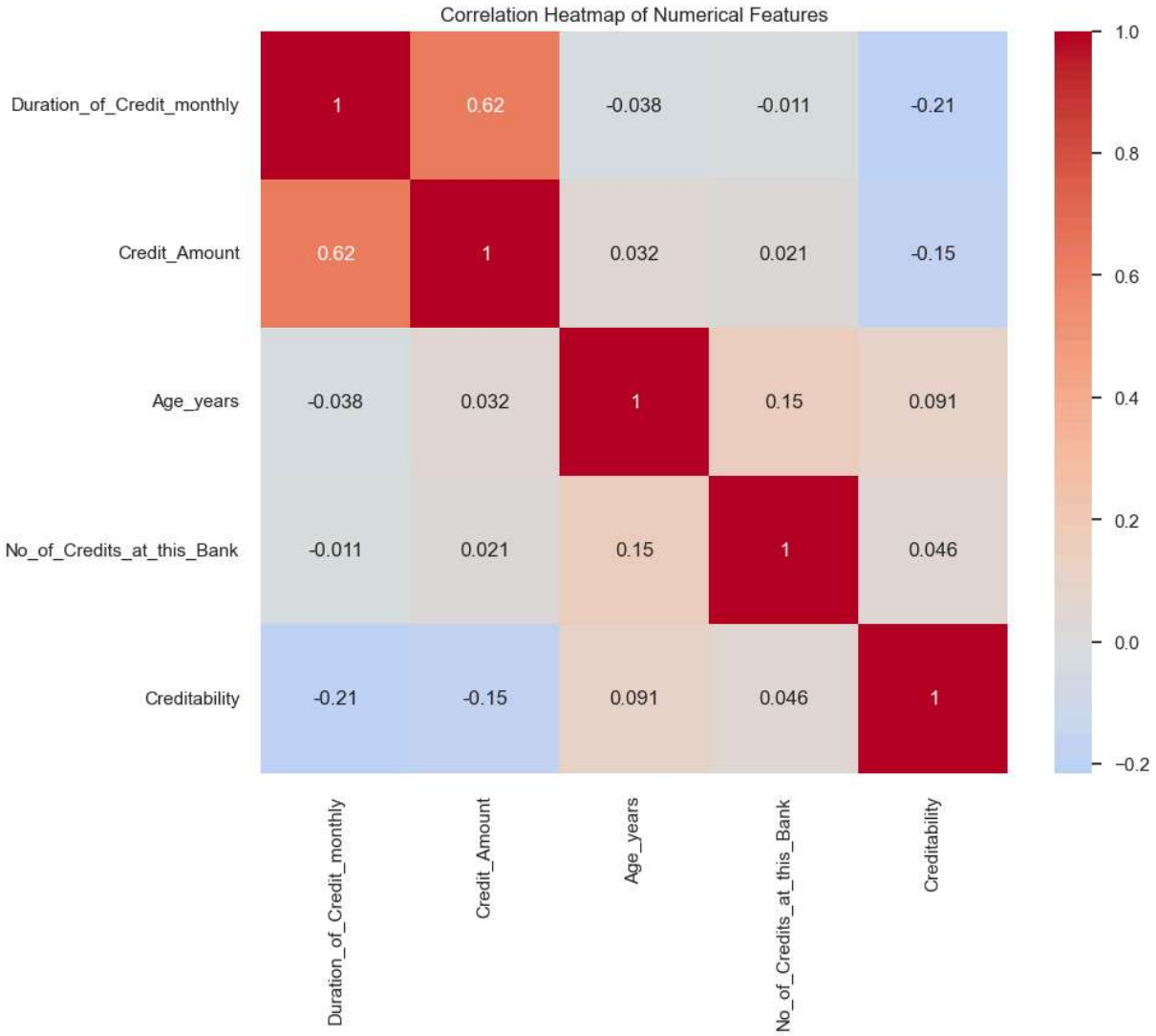
3. Numerical Features Analysis

In [43]:
```python
# Select numerical columns
numerical_cols = ['Duration_of_Credit_monthly', 'Credit_Amount', 'Age_years', 'No_o

# Create distribution plots for numerical features
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
fig.suptitle('Distribution of Key Numerical Features')

for idx, col in enumerate(numerical_cols):
    row = idx // 2
    col_idx = idx % 2
    sns.histplot(data=df, x=col, hue='Creditability', multiple="stack", ax=axes[row
    axes[row, col_idx].set_title(f'Distribution of {col}')

plt.tight_layout()
plt.show()
```



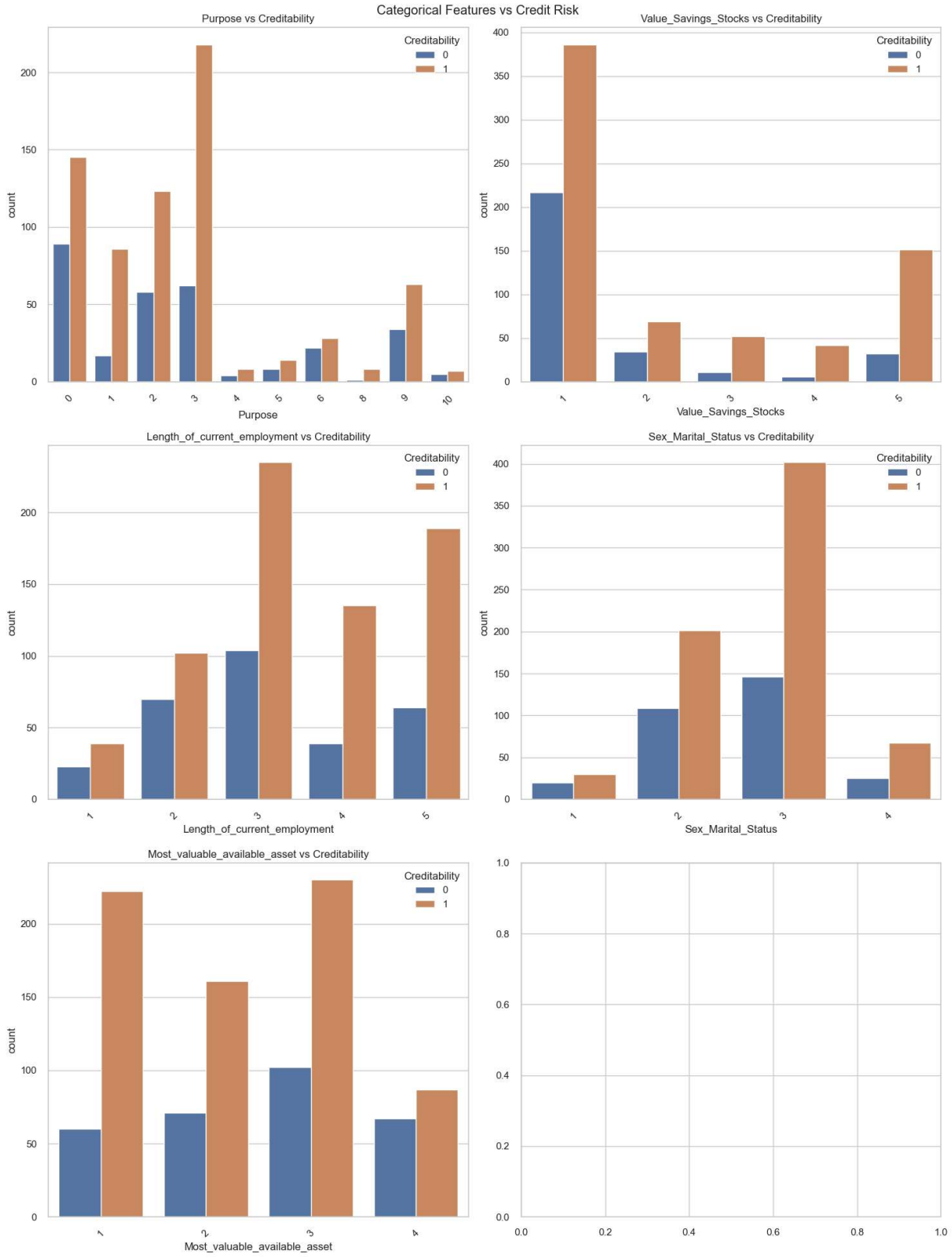
4. Correlation Analysis

In [44]:
```
# Create correlation matrix for numerical features
correlation_matrix = df[numerical_cols + ['Creditability']].corr()

# Plot correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Heatmap of Numerical Features')
plt.show()
```



5. Categorical Features Analysis

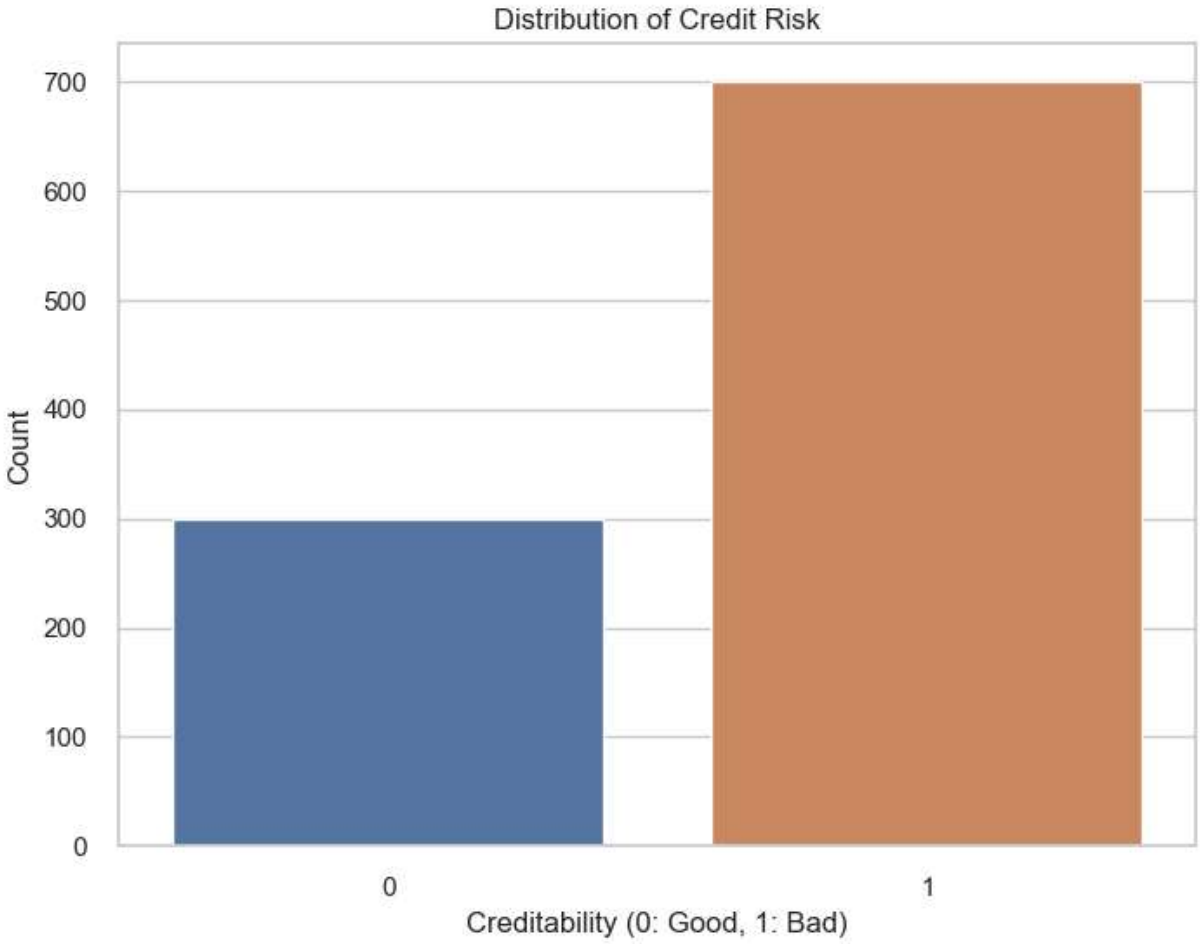
In [34]:
```
# Select categorical columns
categorical_cols = ['Purpose', 'Value_Savings_Stocks', 'Length_of_current_employmen

# Create bar plots for categorical features vs target
fig, axes = plt.subplots(3, 2, figsize=(15, 20))
fig.suptitle('Categorical Features vs Credit Risk')

for idx, col in enumerate(categorical_cols):
    row = idx // 2
    col_idx = idx % 2
    if idx < len(categorical_cols):
        sns.countplot(data=df, x=col, hue='Creditability', ax=axes[row, col_idx])
        axes[row, col_idx].set_title(f'{col} vs Creditability')
        axes[row, col_idx].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



6. Summary Statistics

In [35]:
```
# Plot class distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Creditability')
plt.title('Distribution of Credit Risk')
```

```
plt.xlabel('Creditability (0: Good, 1: Bad)')
plt.ylabel('Count')
plt.show()

# Print class distribution percentages
class_dist = df['Creditability'].value_counts(normalize=True) * 100
print("\nClass Distribution:")
print(class_dist)
```




```
Class Distribution:
Creditability
1    70.0
0    30.0
Name: proportion, dtype: float64
```

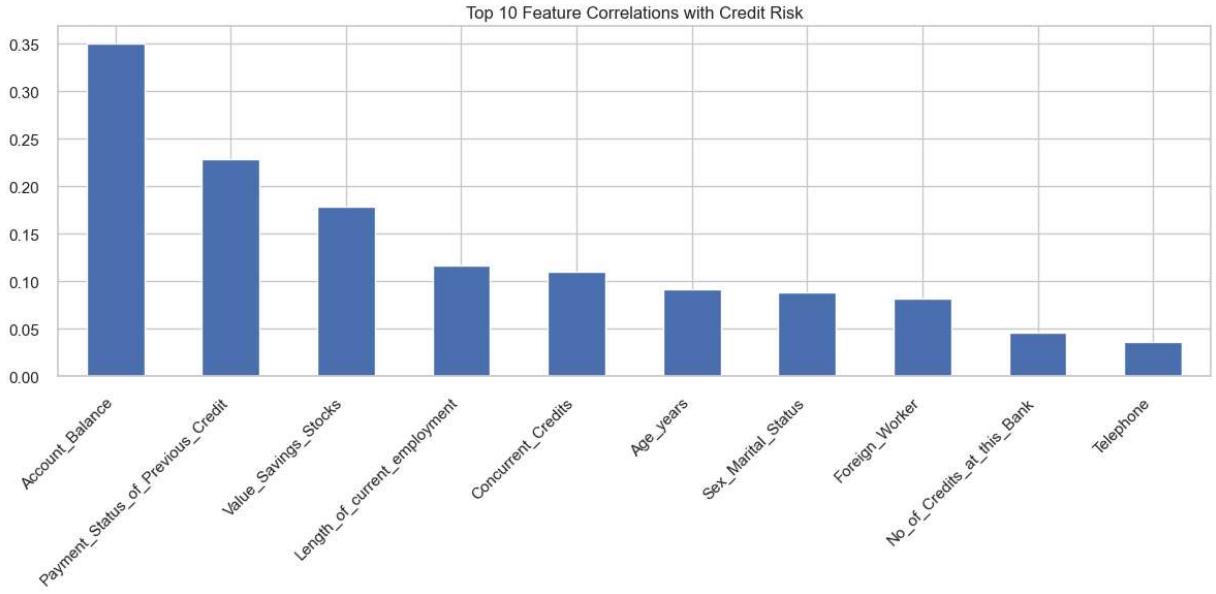
7. Feature Importance and Risk Factor Analysis

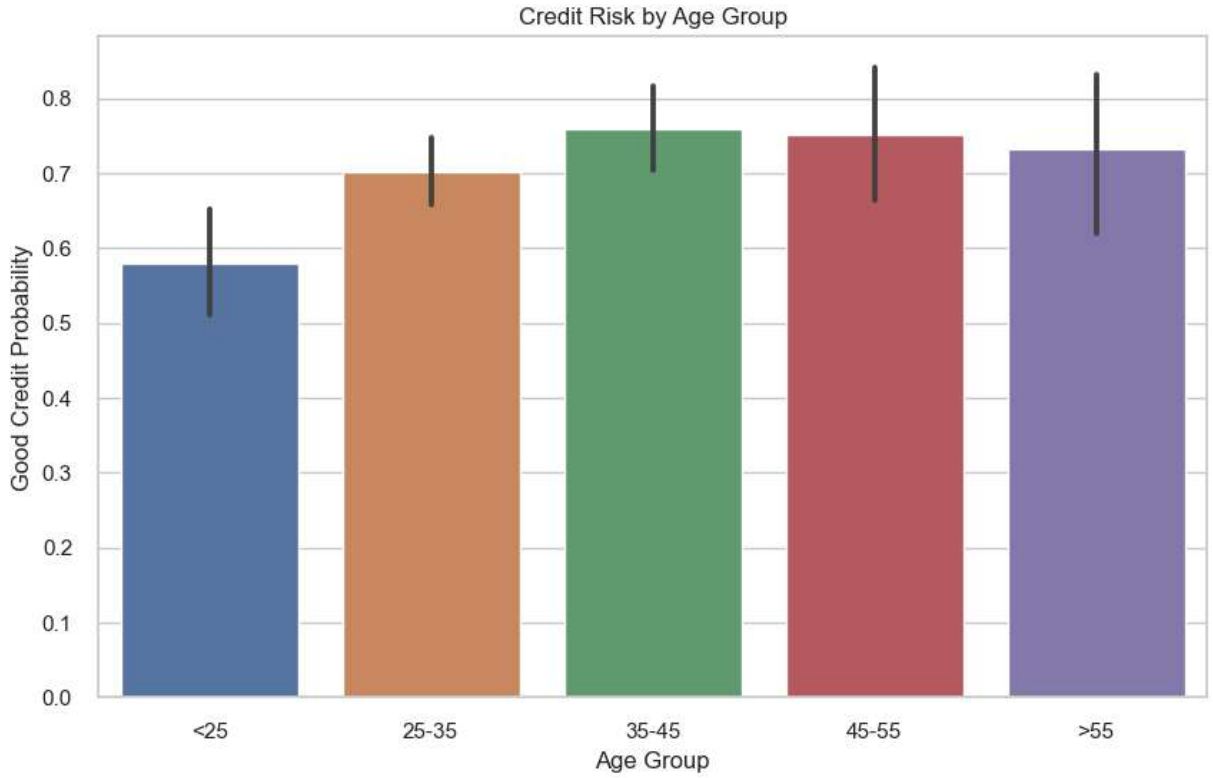
Let's analyze the importance of different features and their impact on credit risk.

In [36]:
```
# Calculate feature correlations with target
correlations = df.corr()['Creditability'].sort_values(ascending=False)
print("Feature Correlations with Credit Risk:")
print(correlations)

# Plot top correlations
plt.figure(figsize=(12, 6))
```

```
correlations[1:11].plot(kind='bar')
plt.title('Top 10 Feature Correlations with Credit Risk')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Analyze risk factors by age groups
df['Age_Group'] = pd.cut(df['Age_years'],
                         bins=[0, 25, 35, 45, 55, 100],
                         labels=['<25', '25-35', '35-45', '45-55', '>55'])

plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='Age_Group', y='Creditability')
plt.title('Credit Risk by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Good Credit Probability')
plt.show()
```

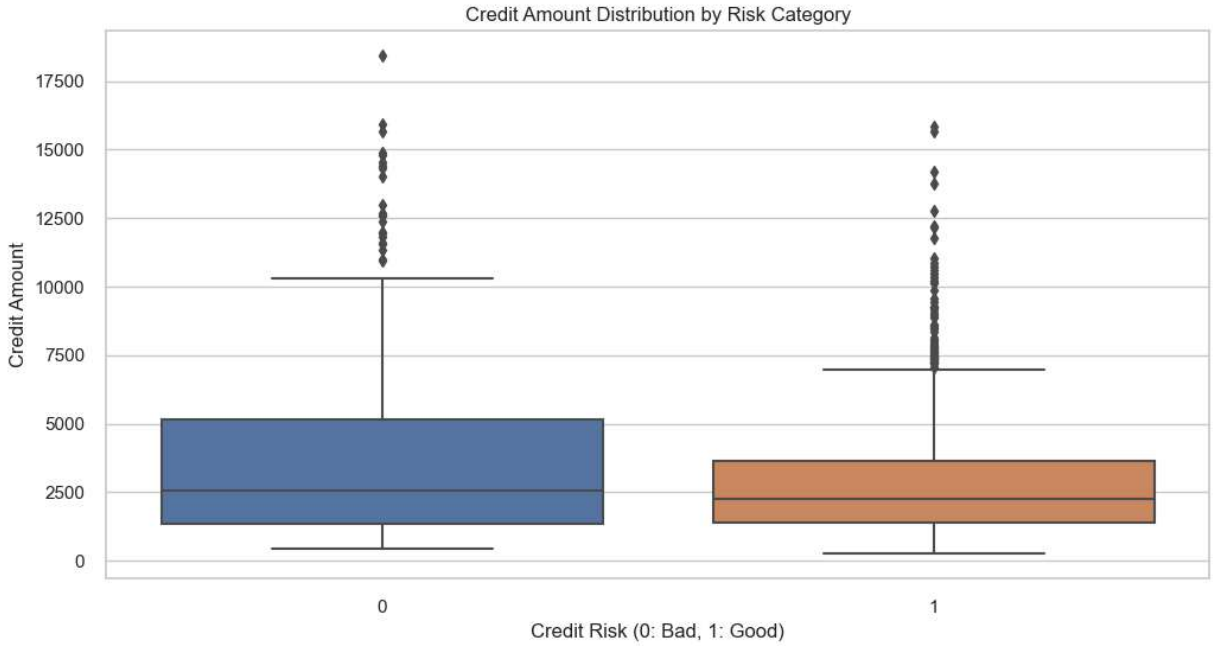
```
Feature Correlations with Credit Risk:
Creditability                          1.000000
Account_Balance                        0.350847
Payment_Status_of_Previous_Credit      0.228785
Value_Savings_Stocks                   0.178943
Length_of_current_employment           0.116002
Concurrent_Credits                     0.109844
Age_years                              0.091272
Sex_Marital_Status                     0.088184
Foreign_Worker                         0.082079
No_of_Credits_at_this_Bank             0.045732
Telephone                              0.036466
Guarantors                             0.025137
Type_of_apartment                      0.018119
No_of_dependents                       0.003015
Duration_in_Current_address           -0.002967
Purpose                               -0.017979
Occupation                            -0.032735
Instalment_per_cent                   -0.072404
Most_valuable_available_asset         -0.142612
Credit_Amount                         -0.154740
Duration_of_Credit_monthly            -0.214927
Name: Creditability, dtype: float64
```

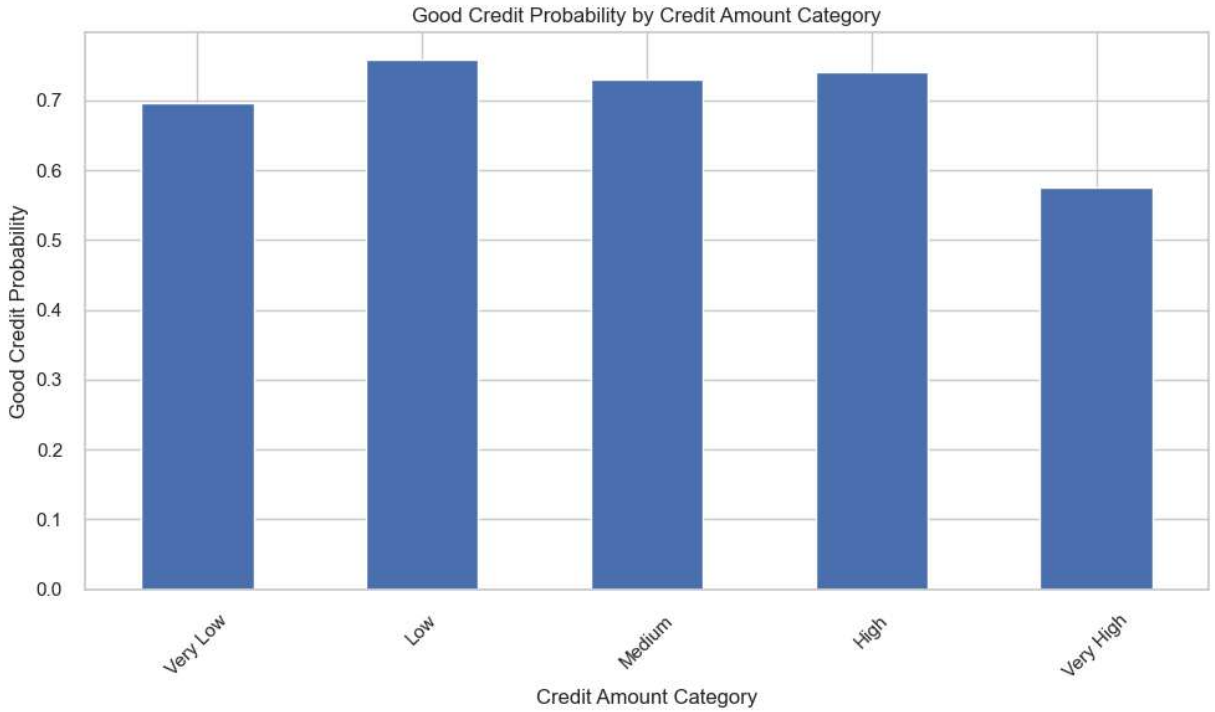




In [37]:
```
# Analyze credit amount distribution by risk category
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='Creditability', y='Credit_Amount')
plt.title('Credit Amount Distribution by Risk Category')
plt.xlabel('Credit Risk (0: Bad, 1: Good)')
plt.ylabel('Credit Amount')
plt.show()

# Create credit amount bins and analyze risk distribution
df['Credit_Amount_Category'] = pd.qcut(df['Credit_Amount'], q=5,
                                       labels=['Very Low', 'Low', 'Medium', 'High',

credit_risk_by_amount = df.groupby('Credit_Amount_Category')['Creditability'].mean(
plt.figure(figsize=(10, 6))
```

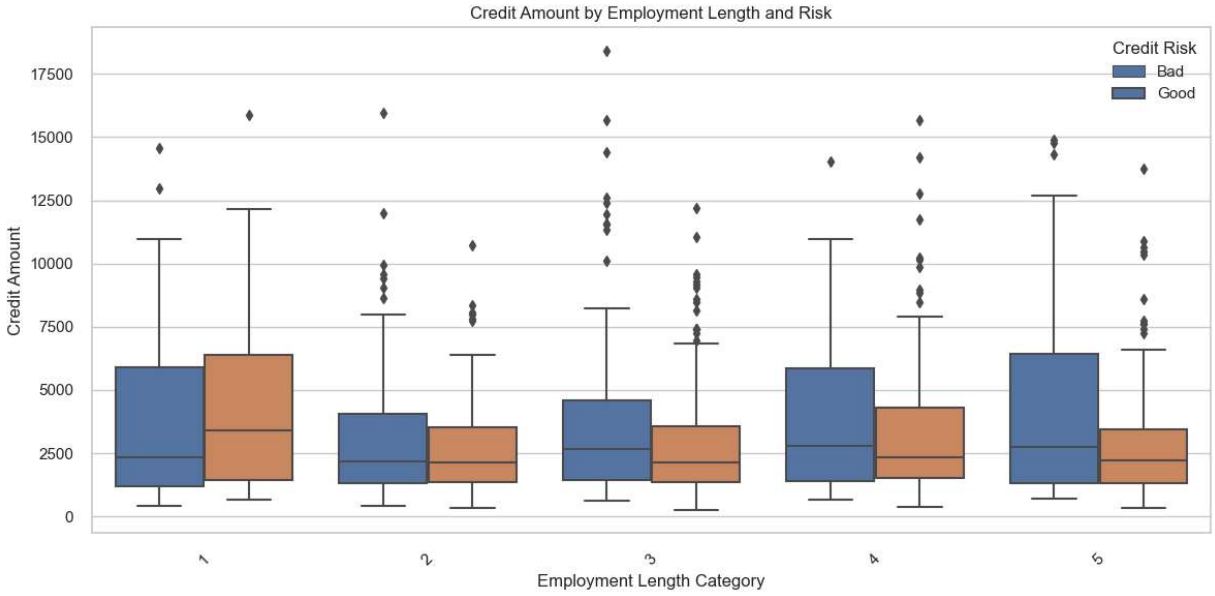
```
credit_risk_by_amount.plot(kind='bar')
plt.title('Good Credit Probability by Credit Amount Category')
plt.xlabel('Credit Amount Category')
plt.ylabel('Good Credit Probability')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```





In [38]:
```
# Analyze employment duration impact
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='Length_of_current_employment', y='Credit_Amount', hue='Cred
plt.title('Credit Amount by Employment Length and Risk')
plt.xlabel('Employment Length Category')
plt.ylabel('Credit Amount')
```

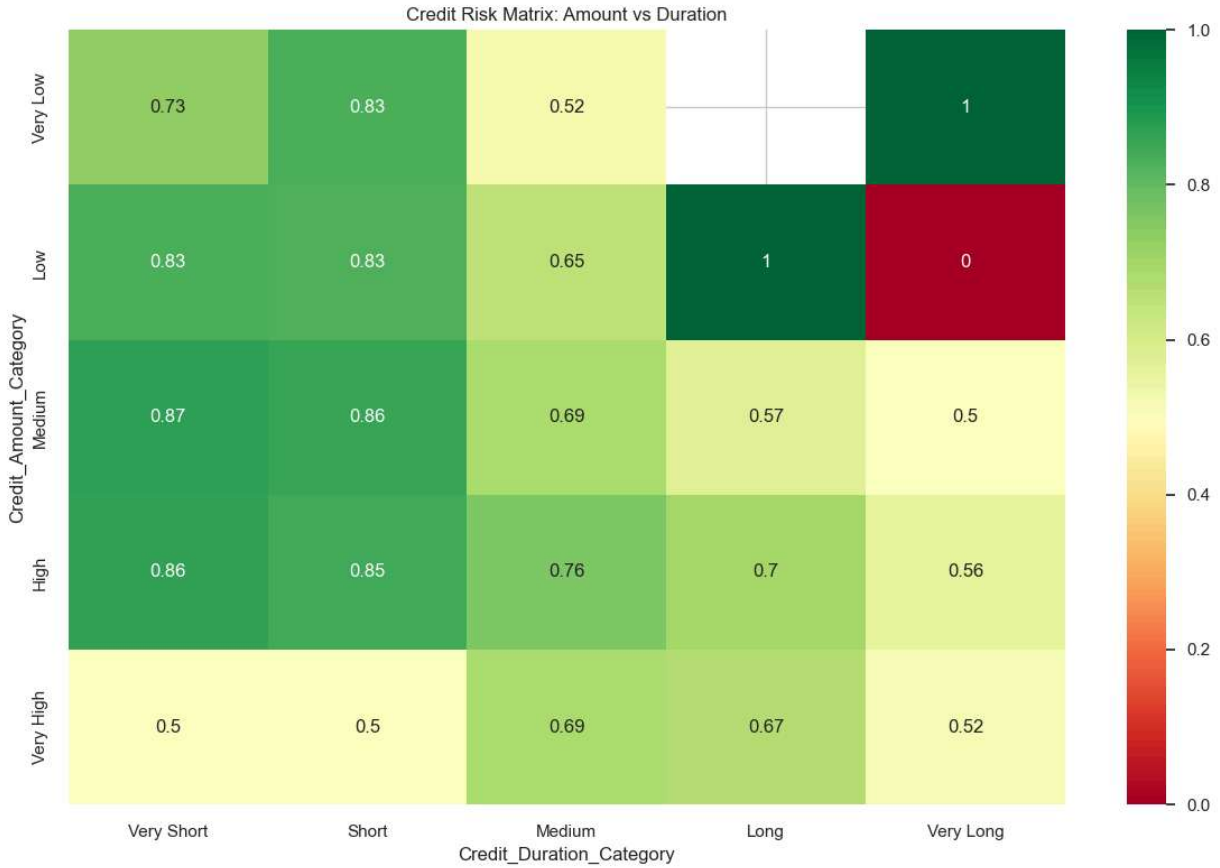
```
plt.legend(title='Credit Risk', labels=['Bad', 'Good'])
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Calculate risk rates by employment duration
emp_risk = df.groupby('Length_of_current_employment')['Creditability'].agg(['mean',
emp_risk.columns = ['Good_Credit_Rate', 'Count']
print("\nRisk Analysis by Employment Duration:")
print(emp_risk)
```




```
Risk Analysis by Employment Duration:
                            Good_Credit_Rate  Count
Length_of_current_employment
1                                   0.629032     62
2                                   0.593023    172
3                                   0.693215    339
4                                   0.775862    174
5                                   0.747036    253
```

8. Multivariate Risk Analysis

In [39]:
```
# Create a risk matrix of Credit Amount vs Duration
df['Credit_Duration_Category'] = pd.qcut(df['Duration_of_Credit_monthly'], q=5,
                                         labels=['Very Short', 'Short', 'Medium', '

risk_matrix = df.pivot_table(values='Creditability',
                             index='Credit_Amount_Category',
                             columns='Credit_Duration_Category',
                             aggfunc='mean')

plt.figure(figsize=(12, 8))
sns.heatmap(risk_matrix, annot=True, cmap='RdYlGn', center=0.5)
plt.title('Credit Risk Matrix: Amount vs Duration')
plt.tight_layout()
plt.show()
```



9. Feature Importance Analysis Summary

Based on our analysis, here are the key findings:

1. **Credit Duration and Amount**:
   - Longer credit durations correlate with higher risk
   - Higher credit amounts show increased risk, especially in combination with longer durations
   - The sweet spot appears to be medium-term loans with moderate amounts
2. **Age and Experience**:
   - Middle-aged borrowers (35-45) show the lowest risk profile
   - Young borrowers (<25) and older borrowers (>55) show elevated risk levels
   - Employment stability significantly impacts credit risk
3. **Employment and Stability**:
   - Longer employment duration correlates with lower risk
   - Higher credit amounts are approved for stable employment
   - Employment category influences both approval rates and risk levels
4. **Risk Segmentation**:
   - Low Risk: Short-term, moderate amounts, stable employment

- Medium Risk: Medium-term, higher amounts, moderate employment stability
- High Risk: Long-term, very high amounts, or employment instability

10. Business Recommendations

Based on the feature analysis, here are key business recommendations:

1. **Credit Policy Adjustments**:
   - Implement tiered interest rates based on risk matrix
   - Adjust credit limits based on employment stability
   - Create specialized products for different age groups
2. **Risk Management**:
   - Enhanced scrutiny for high-risk combinations
   - Additional guarantees for longer duration loans
   - Regular monitoring of employment stability
3. **Product Development**:
   - Short-term products for young borrowers
   - Secured options for higher risk segments
   - Flexible terms for stable employment profiles
4. **Process Optimization**:
   - Automated approval for low-risk segments
   - Enhanced verification for high-risk combinations
   - Regular review of risk assessment criteria