# Conflict & Adaptation Analysis

## How Conflict Handling Relates to Team Outcomes

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

plt.style.use('seaborn-v0_8-whitegrid')
plt.rcParams['figure.figsize'] = (10, 6)
plt.rcParams['font.size'] = 11

# Load data
df = pd.read_excel('SurveyData211.xlsx')

# Key variables
OUTCOME_VARS = ['GO1', 'NPS1', 'NPS2', 'SE1', 'SE2', 'RLS1']
VAR_LABELS = {
    'CA1': 'Conflict Adaptation (1-10)',
    'GO1': 'Growth/Outcomes (1-5)',
    'NPS1': 'Team Satisfaction 1 (1-5)',
    'NPS2': 'Team Satisfaction 2 (1-5)',
    'SE1': 'Self-Efficacy 1 (1-5)',
    'SE2': 'Self-Efficacy 2 (1-5)',
    'RLS1': 'Relationship Strength (1-10)',
    'Section': 'Class Section'
}

print(f"Dataset: {df.shape[0]} responses, {df.shape[1]} variables")
```

```
Dataset: 210 responses, 27 variables
```

## . Distribution of Conflict Adaptation Scores

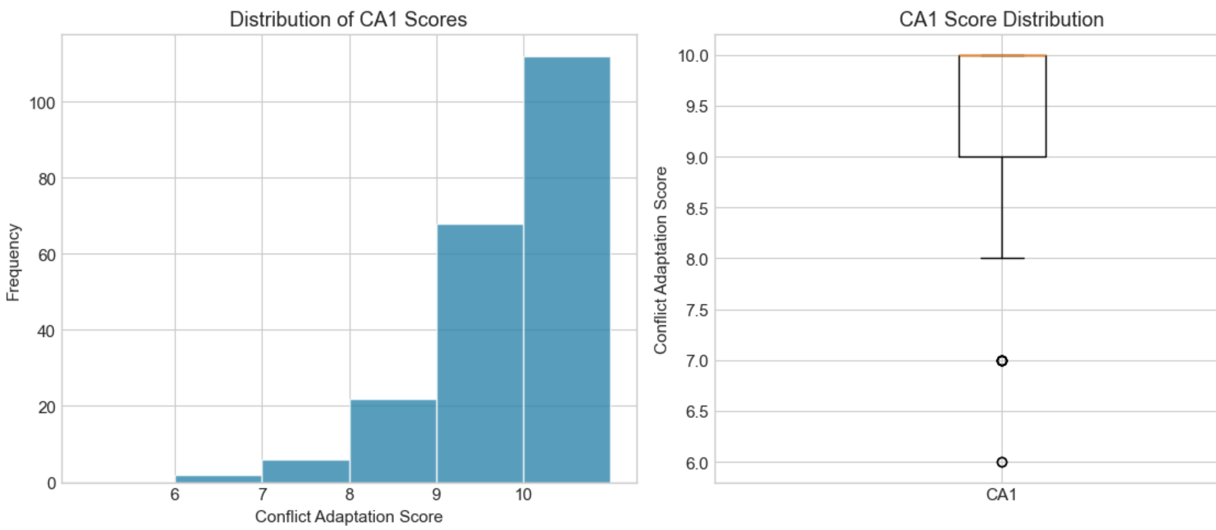
In [2]:
```
# Summary statistics for CA1 and outcome variables
summary_vars = ['CA1'] + OUTCOME_VARS
summary_stats = df[summary_vars].describe().round(2)
summary_stats
```

Out[2]:

| | CA | GO | NPS | NPS | SE | SE | RLS |
|---|---|---|---|---|---|---|---|
| **count** | . | . | . | . | . | . | . |
| **mean** | . | . | . | . | . | . | . |
| **std** | . | . | . | . | . | . | . |
| **min** | . | . | . | . | . | . | . |
| **%** | . | . | . | . | . | . | . |
| **%** | . | . | . | . | . | . | . |
| **%** | . | . | . | . | . | . | . |
| **max** | . | . | . | . | . | . | . |

In [3]:

```
# Figure 1: Distribution of Conflict Adaptation Scores
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Histogram
axes[0].hist(df['CA1'].dropna(), bins=range(5, 12), edgecolor='white', co
axes[0].set_xlabel('Conflict Adaptation Score')
axes[0].set_ylabel('Frequency')
axes[0].set_title('Distribution of CA1 Scores')
axes[0].set_xticks(range(6, 11))

# Box plot
axes[1].boxplot(df['CA1'].dropna(), vert=True)
axes[1].set_ylabel('Conflict Adaptation Score')
axes[1].set_title('CA1 Score Distribution')
axes[1].set_xticklabels(['CA1'])

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

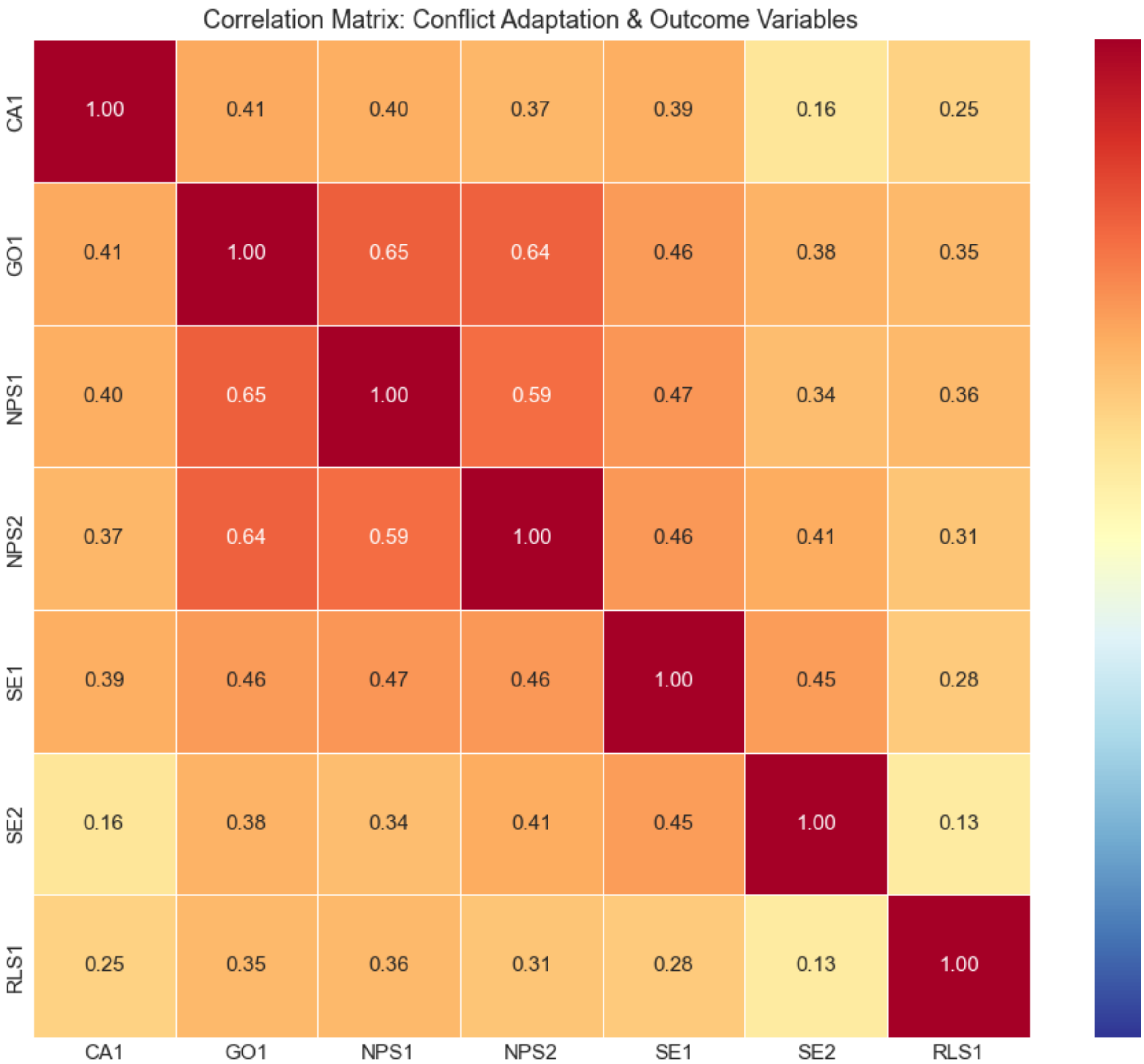



## . Correlation Analysis: CA vs Outcome Variables

In [4]:

```
# Figure 2: Correlation Heatmap - CA1 vs All Outcome Variables
corr_vars = ['CA1'] + OUTCOME_VARS
corr_matrix = df[corr_vars].corr().round(3)
```

```
fig, ax = plt.subplots(figsize=(10, 8))
mask = np.triu(np.ones_like(corr_matrix, dtype=bool), k=1)
sns.heatmap(corr_matrix, annot=True, cmap='RdYlBu_r', center=0,
            vmin=-1, vmax=1, square=True, linewidths=0.5, ax=ax,
            fmt='.2f', annot_kws={'size': 11})
ax.set_title('Correlation Matrix: Conflict Adaptation & Outcome Variables
plt.tight_layout()
plt.show()
```




In [5]:
```
# Table: CA1 correlations with outcome variables
ca_corrs = corr_matrix['CA1'].drop('CA1').sort_values(ascending=False)
corr_table = pd.DataFrame({
    'Variable': ca_corrs.index,
    'Correlation with CA1': ca_corrs.values,
    'Interpretation': ['Strong positive' if abs(r) > 0.5 else 'Moderate p
                       else 'Weak positive' if r > 0 else 'Weak negative'
})
corr_table
```

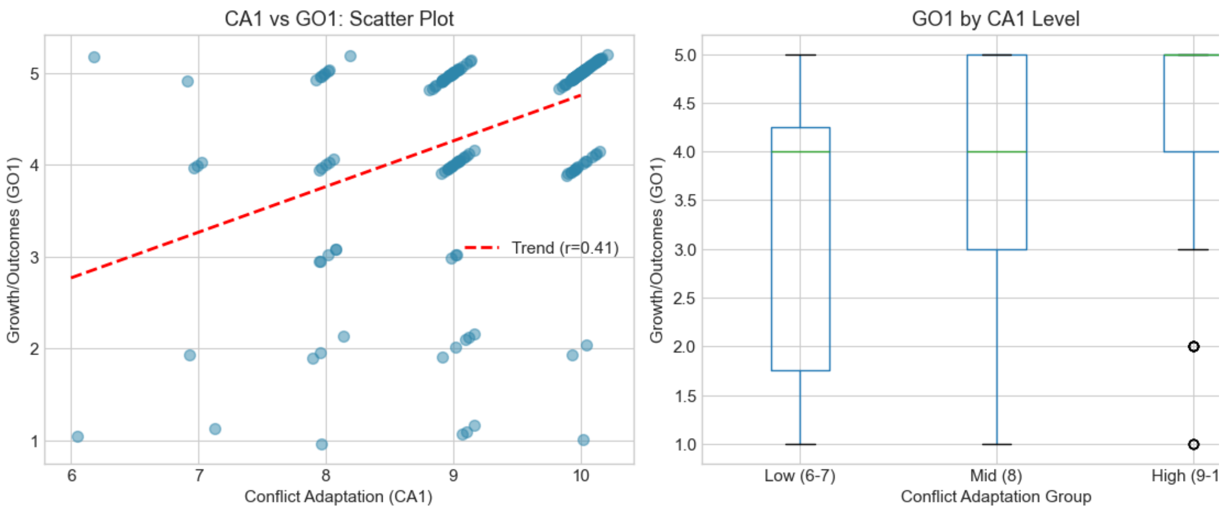
Out[5]:

| Variable | Correlation with CA | Interpretation |
|---|---|---|
| GO | . | Moderate positive |
| NPS | . | Moderate positive |
| SE | . | Moderate positive |
| NPS | . | Moderate positive |
| RLS | . | Weak positive |
| SE | . | Weak positive |

## . CA vs Outcome Variables: Detailed Relationship

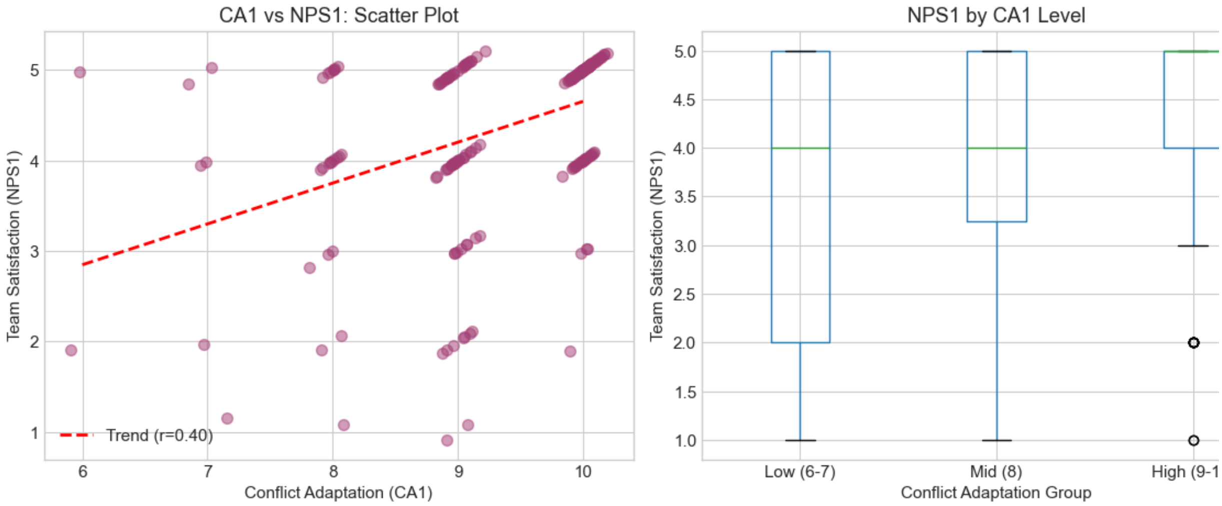
In [6]:

```
# Figure 3: CA1 vs Growth/Outcomes (GO1)
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Scatter with jitter
jitter = np.random.normal(0, 0.08, len(df))
axes[0].scatter(df['CA1'] + jitter, df['GO1'] + jitter, alpha=0.5, c='#2E
z = np.polyfit(df['CA1'].dropna(), df.loc[df['CA1'].notna(), 'GO1'], 1)
p = np.poly1d(z)
x_line = np.linspace(df['CA1'].min(), df['CA1'].max(), 100)
axes[0].plot(x_line, p(x_line), 'r--', linewidth=2, label=f'Trend (r={cor
axes[0].set_xlabel('Conflict Adaptation (CA1)')
axes[0].set_ylabel('Growth/Outcomes (GO1)')
axes[0].set_title('CA1 vs GO1: Scatter Plot')
axes[0].legend()

# Box plot by CA1 groups
df['CA1_group'] = pd.cut(df['CA1'], bins=[5, 7, 8, 10], labels=['Low (6-7
df.boxplot(column='GO1', by='CA1_group', ax=axes[1])
axes[1].set_xlabel('Conflict Adaptation Group')
axes[1].set_ylabel('Growth/Outcomes (GO1)')
axes[1].set_title('GO1 by CA1 Level')
plt.suptitle('')

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



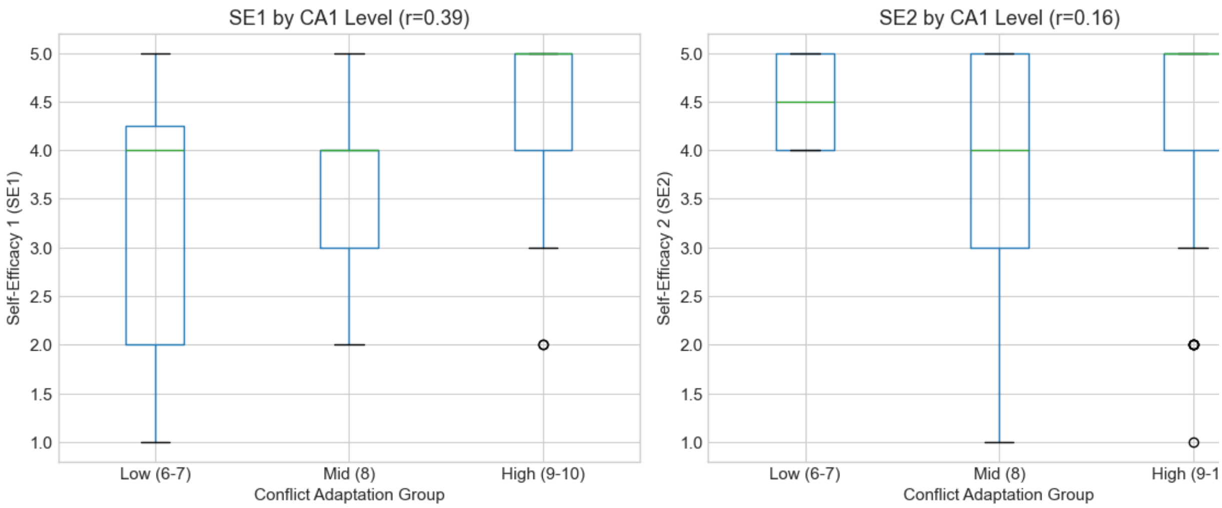
In [7]:

```
# Figure 4: CA1 vs Team Satisfaction (NPS1)
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Scatter
jitter = np.random.normal(0, 0.08, len(df))
axes[0].scatter(df['CA1'] + jitter, df['NPS1'] + jitter, alpha=0.5, c='#A
z = np.polyfit(df['CA1'].dropna(), df.loc[df['CA1'].notna(), 'NPS1'], 1)
p = np.poly1d(z)
axes[0].plot(x_line, p(x_line), 'r--', linewidth=2, label=f'Trend (r={cor
axes[0].set_xlabel('Conflict Adaptation (CA1)')
axes[0].set_ylabel('Team Satisfaction (NPS1)')
axes[0].set_title('CA1 vs NPS1: Scatter Plot')
axes[0].legend()

# Box plot
df.boxplot(column='NPS1', by='CA1_group', ax=axes[1])
axes[1].set_xlabel('Conflict Adaptation Group')
axes[1].set_ylabel('Team Satisfaction (NPS1)')
axes[1].set_title('NPS1 by CA1 Level')
plt.suptitle('')

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



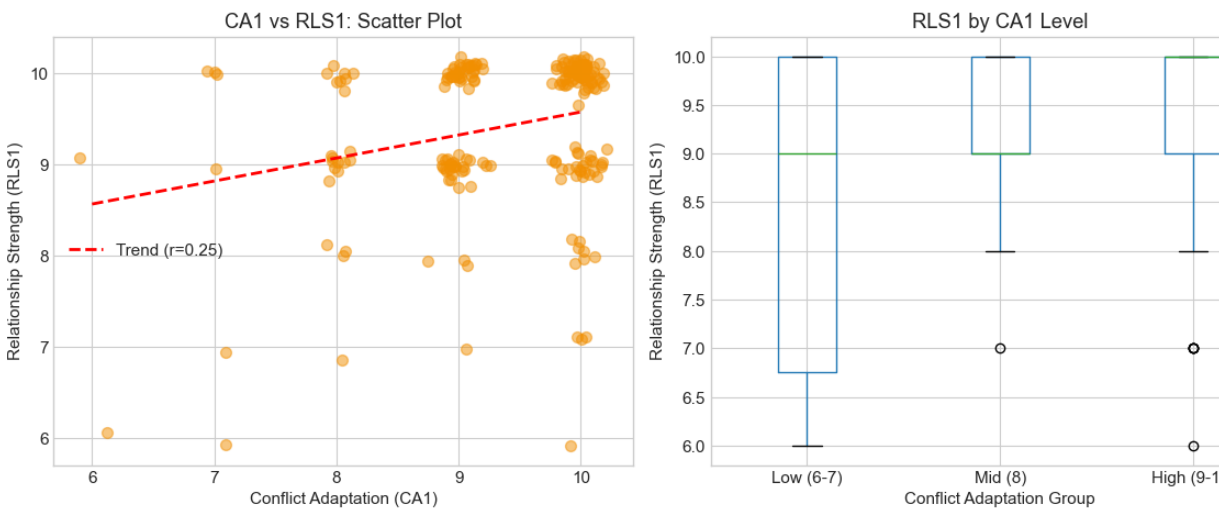

In [8]:

```
# Figure 5: CA1 vs Self-Efficacy (SE1 and SE2)
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# SE1 boxplot by CA1 group
df.boxplot(column='SE1', by='CA1_group', ax=axes[0])
axes[0].set_xlabel('Conflict Adaptation Group')
axes[0].set_ylabel('Self-Efficacy 1 (SE1)')
axes[0].set_title(f'SE1 by CA1 Level (r={corr_matrix.loc["CA1","SE1"]:.2f
plt.suptitle('')

# SE2 boxplot by CA1 group
df.boxplot(column='SE2', by='CA1_group', ax=axes[1])
axes[1].set_xlabel('Conflict Adaptation Group')
axes[1].set_ylabel('Self-Efficacy 2 (SE2)')
axes[1].set_title(f'SE2 by CA1 Level (r={corr_matrix.loc["CA1","SE2"]:.2f
plt.suptitle('')

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




In [9]:

```
# Figure 6: CA1 vs Relationship Strength (RLS1)
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Scatter
jitter_x = np.random.normal(0, 0.1, len(df))
jitter_y = np.random.normal(0, 0.1, len(df))
axes[0].scatter(df['CA1'] + jitter_x, df['RLS1'] + jitter_y, alpha=0.5, c
z = np.polyfit(df['CA1'].dropna(), df.loc[df['CA1'].notna(), 'RLS1'], 1)
p = np.poly1d(z)
axes[0].plot(x_line, p(x_line), 'r--', linewidth=2, label=f'Trend (r={cor
axes[0].set_xlabel('Conflict Adaptation (CA1)')
axes[0].set_ylabel('Relationship Strength (RLS1)')
axes[0].set_title('CA1 vs RLS1: Scatter Plot')
axes[0].legend()

# Box plot
df.boxplot(column='RLS1', by='CA1_group', ax=axes[1])
axes[1].set_xlabel('Conflict Adaptation Group')
axes[1].set_ylabel('Relationship Strength (RLS1)')
axes[1].set_title('RLS1 by CA1 Level')
plt.suptitle('')

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




## . High vs Low Conflict Adaptation: Group Comparison

In [10]:
```
# Split by median CA1 score
ca_median = df['CA1'].median()
df['CA1_binary'] = df['CA1'].apply(lambda x: 'High CA' if x >= ca_median

# Comparison table
comparison = df.groupby('CA1_binary')[OUTCOME_VARS].agg(['mean', 'std']).
comparison.columns = [f'{col[0]}_{col[1]}' for col in comparison.columns]

# Reshape for cleaner display
comparison_clean = pd.DataFrame({
    'Group': ['High CA (≥{:.0f})'.format(ca_median), 'Low CA (<{:.0f})'.f
    'N': [len(df[df['CA1_binary'] == 'High CA']), len(df[df['CA1_binary']
    'GO1 Mean': [comparison.loc['High CA', 'GO1_mean'], comparison.loc['L
    'NPS1 Mean': [comparison.loc['High CA', 'NPS1_mean'], comparison.loc[
    'SE1 Mean': [comparison.loc['High CA', 'SE1_mean'], comparison.loc['L
    'RLS1 Mean': [comparison.loc['High CA', 'RLS1_mean'], comparison.loc[
})
print(f"Median CA1 score: {ca_median}")
comparison_clean
```

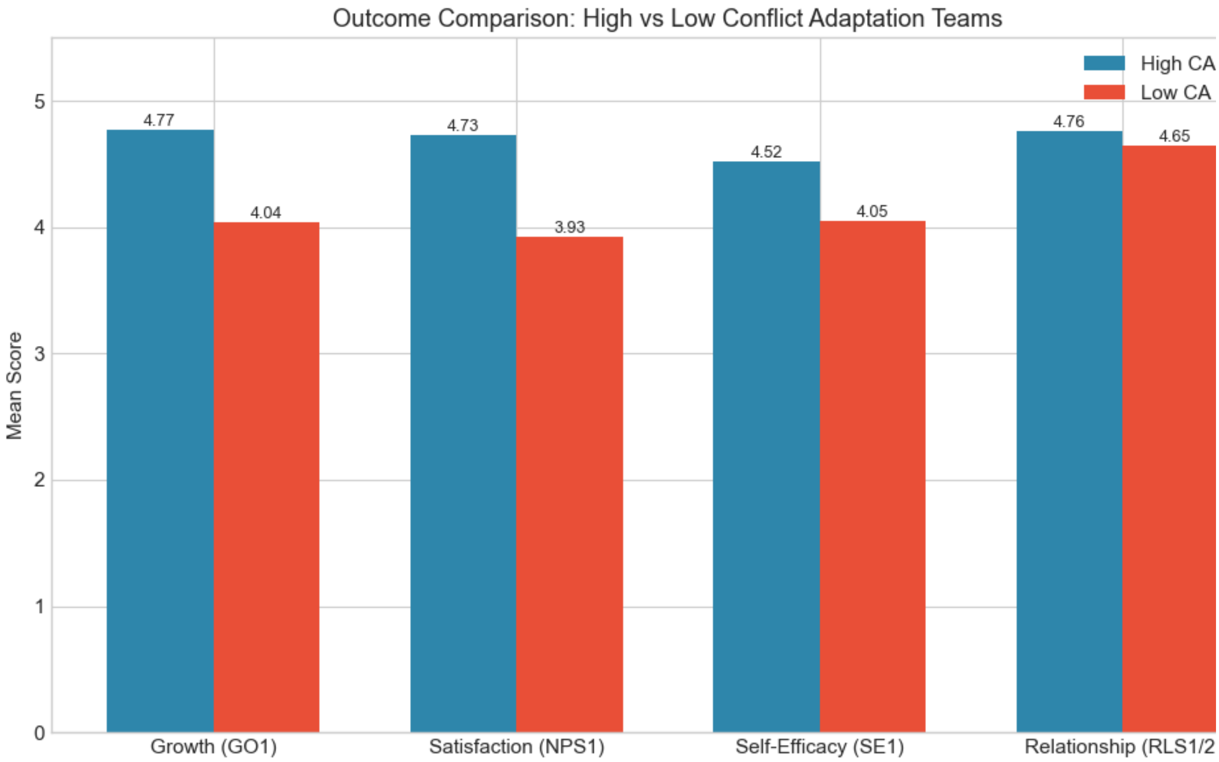
```
Median CA1 score: 10.0
```

Out[10]:

| Group | N | GO | Mean | NPS | Mean | SE | Mean | RLS | Mean |
|---|---|---|---|---|---|---|---|---|---|
| High CA (≥ ) | | | . | | . | | . | | . |
| Low CA (< ) | | | . | | . | | . | | . |

In [11]:
```
# Figure 7: High vs Low CA Comparison Bar Chart
fig, ax = plt.subplots(figsize=(10, 6))

outcome_labels = ['Growth (GO1)', 'Satisfaction (NPS1)', 'Self-Efficacy (
high_ca_means = [
    df[df['CA1_binary'] == 'High CA']['GO1'].mean(),
    df[df['CA1_binary'] == 'High CA']['NPS1'].mean(),
    df[df['CA1_binary'] == 'High CA']['SE1'].mean(),
    df[df['CA1_binary'] == 'High CA']['RLS1'].mean() / 2  # Scale to 1-5
]
low_ca_means = [
    df[df['CA1_binary'] == 'Low CA']['GO1'].mean(),
    df[df['CA1_binary'] == 'Low CA']['NPS1'].mean(),
    df[df['CA1_binary'] == 'Low CA']['SE1'].mean(),
    df[df['CA1_binary'] == 'Low CA']['RLS1'].mean() / 2  # Scale to 1-5
]

x = np.arange(len(outcome_labels))
width = 0.35

bars1 = ax.bar(x - width/2, high_ca_means, width, label=f'High CA (≥{ca_m
bars2 = ax.bar(x + width/2, low_ca_means, width, label=f'Low CA (<{ca_med

ax.set_ylabel('Mean Score')
ax.set_title('Outcome Comparison: High vs Low Conflict Adaptation Teams')
ax.set_xticks(x)
ax.set_xticklabels(outcome_labels)
ax.legend()
ax.set_ylim(0, 5.5)

# Add value labels
```

```
for bar in bars1:
    ax.annotate(f'{bar.get_height():.2f}', xy=(bar.get_x() + bar.get_widt
                ha='center', va='bottom', fontsize=9)
for bar in bars2:
    ax.annotate(f'{bar.get_height():.2f}', xy=(bar.get_x() + bar.get_widt
                ha='center', va='bottom', fontsize=9)

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




## . Conflict Adaptation by Section

In [12]:

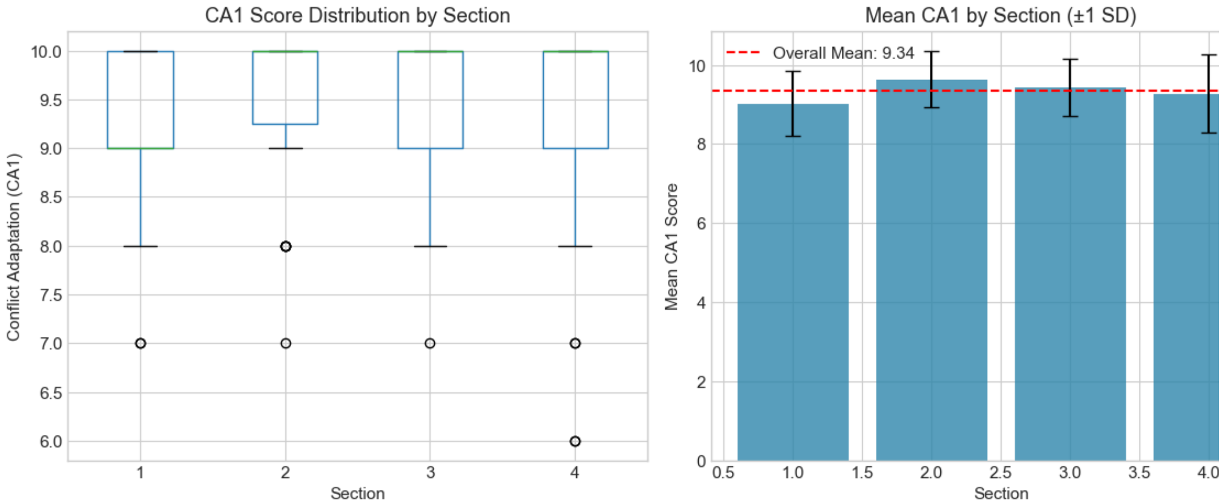
```
# Table: CA1 and outcomes by section
section_stats = df.groupby('Section').agg({
    'CA1': ['mean', 'std', 'count'],
    'GO1': 'mean',
    'NPS1': 'mean',
    'RLS1': 'mean'
}).round(2)
section_stats.columns = ['CA1 Mean', 'CA1 Std', 'N', 'GO1 Mean', 'NPS1 Me
section_stats
```

Out[12]:

| | CA | Mean | CA | Std | N | GO | Mean | NPS | Mean | RLS | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Section** | | | | | | | | | | | |
| | | . | | . | | | . | | . | | . |
| | | . | | . | | | . | | . | | . |
| | | . | | . | | | . | | . | | . |
| | | . | | . | | | . | | . | | . |

In [13]:

```
# Figure 8: CA1 Distribution by Section
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Box plot
df.boxplot(column='CA1', by='Section', ax=axes[0])
axes[0].set_xlabel('Section')
axes[0].set_ylabel('Conflict Adaptation (CA1)')
axes[0].set_title('CA1 Score Distribution by Section')
plt.suptitle('')

# Bar chart of means with error bars
section_means = df.groupby('Section')['CA1'].mean()
section_stds = df.groupby('Section')['CA1'].std()
sections = section_means.index

axes[1].bar(sections, section_means, yerr=section_stds, capsize=5, color=
axes[1].set_xlabel('Section')
axes[1].set_ylabel('Mean CA1 Score')
axes[1].set_title('Mean CA1 by Section (±1 SD)')
axes[1].axhline(y=df['CA1'].mean(), color='red', linestyle='--', label=f'
axes[1].legend()

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




# Summary of Key Findings

**Key observations from Conflict Adaptation (CA ) analysis:**

. **Distribution**: Most teams report moderate-to-high conflict adaptation scores (median aroun
- / )

. **Correlations**: CA shows positive correlations with all outcome variables - teams that ha
conflict well tend to report better outcomes

. **Strongest relationships**:

- CA correlates most strongly with RLS (relationship strength) and NPS (tean satisfaction)
- Constructive conflict handling appears linked to stronger interpersonal bonds

. **High vs Low CA**: Teams with above-median CA scores show consistently higher means all outcome metrics

. **Section differences**: Some variation exists between sections in conflict adaptation patterns