# Step 10

## Evaluation & Monitoring

Continuous Performance Tracking & Optimization

McDonald's Market Segmentation Analysis

---

Closing the Loop

November 9, 2025

# Step 10: Evaluation & Monitoring Continuous Performance Tracking & Optimization

## McDonald's Market Segmentation Analysis

Closing the Loop

November 9, 2025

**Abstract**

Step 10 completes the market segmentation framework by establishing systematic evaluation and monitoring mechanisms to track performance, measure success, and enable continuous optimization. This final step transforms the segmentation strategy from a one-time analysis into an ongoing management system through four core components: (1) Performance Measurement Framework defining KPIs across segment-level, campaign-level, and business-level metrics; (2) Data Collection & Tracking Infrastructure specifying tools, frequencies, and responsibilities; (3) Analysis & Reporting Protocol establishing dashboards, review cycles, and decision triggers; and (4) Continuous Improvement Process enabling adaptive strategy refinement. The framework monitors McDonald's three target segments (Segment 2: Happy Value Hunters, Segment 4: Premium Experience Seekers, Segment 1: Price-Quality Skeptics) across the 18-month implementation roadmap, with quarterly business reviews, monthly operational reviews, and real-time performance dashboards ensuring accountability and agility.

## Contents

# 1   The Evaluation Challenge

> ### Why Monitor & Evaluate?
>
> Steps 1-9 designed and launched the segmentation strategy. Step 10 ensures it works.
>
> **Critical Questions:**
>
> - Are we reaching target segments effectively?
>
> - Are segment perceptions shifting as intended?
>
> - Are marketing programs delivering ROI?
>
> - Do segments remain stable over time?
>
> - Should we adjust strategies based on performance?
>
> **Without monitoring:**
>
> - Investments continue despite poor performance
>
> - Opportunities for optimization are missed
>
> - Segment evolution goes undetected
>
> - Accountability is impossible
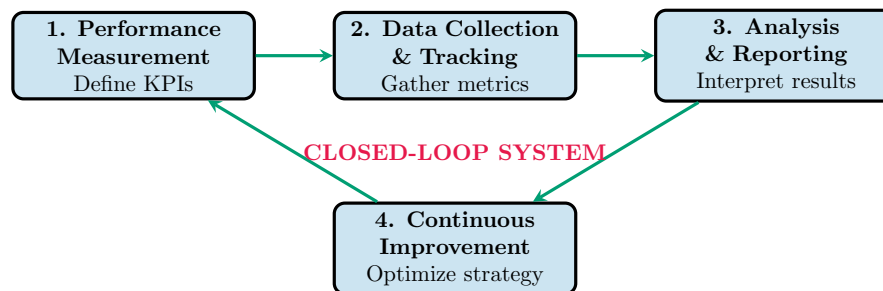
## 1.1   The Monitoring Framework Overview



Figure 1: Four-Component Monitoring Framework

# 2 Component 1: Performance Measurement Framework

## 2.1 Three-Level KPI Hierarchy
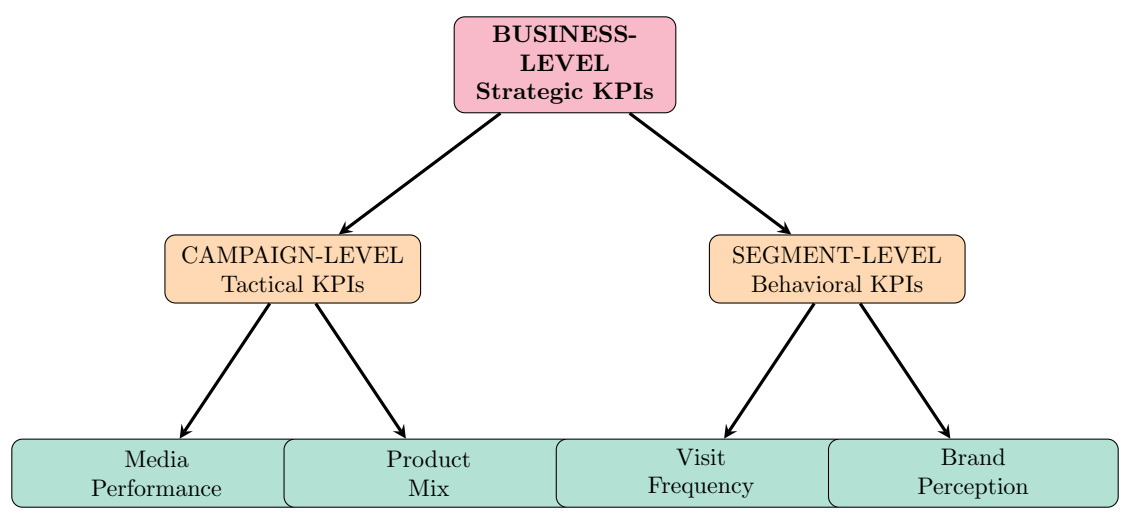


Figure 2: Three-Level KPI Hierarchy for Segmentation Strategy

## 2.2 Detailed KPI Specifications

Table 1: Business-Level KPIs (Strategic)

| KPI | Definition | Target | Frequency | Owner |
|-----|-----------|--------|-----------|-------|
| Total Revenue Growth | YoY revenue increase | +8% | Quarterly | CFO |
| Target Segment Revenue | Revenue from Seg 2,4,1 | 78% of total | Quarterly | CMO |
| Market Share | Share within target demos | +2 pts | Quarterly | CMO |
| Customer Lifetime Value | Avg CLV by segment | +15% | Quarterly | CFO |
| Overall Brand Health | Composite brand metrics | 75+ score | Quarterly | CMO |

Table 2: Campaign-Level KPIs (Tactical)

| KPI | Definition | Target | Frequency | Owner |
|-----|-----------|--------|-----------|-------|
| Campaign ROI | Revenue / Marketing Spend | 3:1 | Monthly | Marketing |
| Cost Per Acquisition | Cost to acquire new customer | ¡$15 | Monthly | Marketing |
| Media Efficiency | Impressions / Dollar | Varies | Monthly | Media Team |
| Digital Engagement | App downloads, usage | +25% QoQ | Monthly | Digital Team |
| Promotion Redemption | Coupon/offer usage rate | ¿40% | Weekly | Operations |

5

Table 3: Segment-Level KPIs (Behavioral)

| KPI | Segment 2 | Segment 4 | Segment 1 | Frequency |
|-----|-----------|-----------|-----------|-----------|
| Visit Frequency | 2.94 → 3.5 | 2.86 → 3.2 | 1.40 → 2.0 | Monthly |
| Like Rating | +2.72 (maintain) | +2.31 (maintain) | +1.03 → +1.5 | Quarterly |
| Transaction Size | $7 (maintain) | $7 → $10 | $6 → $7 | Monthly |
| Loyalty Enrollment | 60% enrolled | N/A | N/A | Monthly |
| Premium Trial | N/A | 40% try Signature | N/A | Quarterly |
| Perception Shift | Maintain "cheap" | Maintain "quality" | 90% → 75% "expensive" | Quarterly |

# 3    Component 2: Data Collection & Tracking Infrastructure

## 3.1    Data Sources & Tools
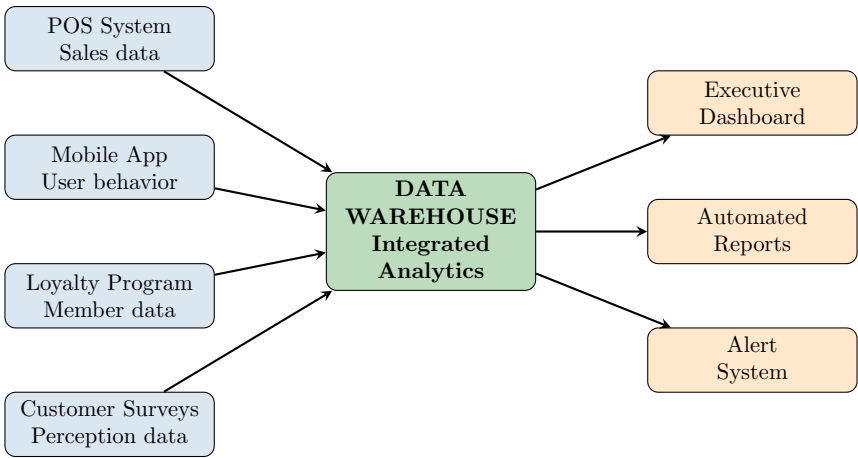


Figure 3: Data Collection Infrastructure

## 3.2    Data Collection Schedule

Table 4: Data Collection Frequency Matrix

| Data Type | Source | Frequency | Purpose |
|-----------|--------|-----------|---------|
| Transaction Data | POS System | Real-time | Visit frequency, basket size |
| App Usage | Mobile App | Real-time | Digital engagement, ordering |
| Loyalty Activity | Rewards Program | Daily | Member behavior, retention |
| Social Media | Social Listening | Daily | Sentiment, brand mentions |
| Media Performance | Ad Platforms | Weekly | Campaign effectiveness |
| Customer Surveys | Online/In-store | Monthly | Perception, satisfaction |
| Brand Tracking | Market Research | Quarterly | Brand health, awareness |
| Competitive Intel | Third-party | Quarterly | Market positioning |

# 4 Component 3: Analysis & Reporting Protocol

## 4.1 Performance Dashboard Architecture

**MCDONALD'S SEGMENTATION
PERFORMANCE DASHBOARD**

**EXECUTIVE SUMMARY:** Overall performance vs. targets
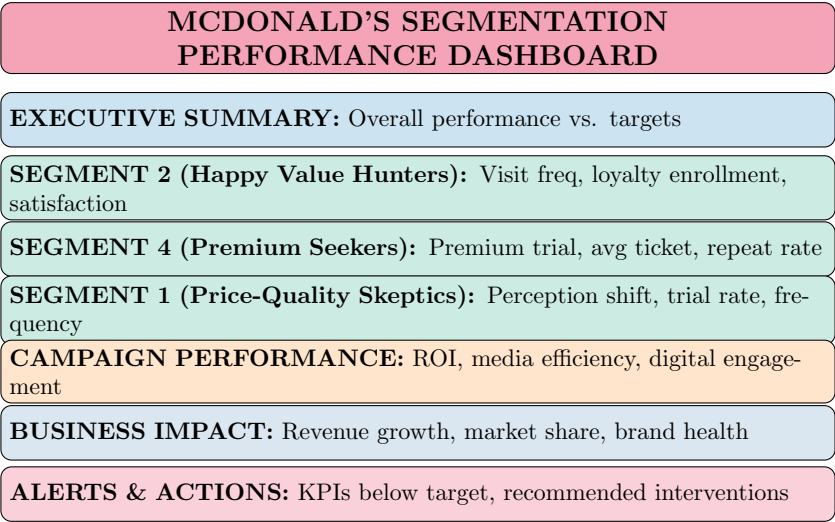
**SEGMENT 2 (Happy Value Hunters):** Visit freq, loyalty enrollment, satisfaction

**SEGMENT 4 (Premium Seekers):** Premium trial, avg ticket, repeat rate

**SEGMENT 1 (Price-Quality Skeptics):** Perception shift, trial rate, frequency

**CAMPAIGN PERFORMANCE:** ROI, media efficiency, digital engagement

**BUSINESS IMPACT:** Revenue growth, market share, brand health

**ALERTS & ACTIONS:** KPIs below target, recommended interventions

Figure 4: Executive Dashboard Structure

## 4.2 Reporting Cycle & Governance

**WEEKLY
Operations Review
1 hour**
aggregate →
**MONTHLY
Tactical Review
2 hours**
synthesize →
**QUARTERLY
Strategic Review
4 hours**

Ops team
KPI check
Quick fixes
reset

Marketing team
Campaign performance
Tactical adjustments

Executive team
strategic Strategic assessment
Major decisions

**ANNUAL
Segmentation
Refresh
Full day**

Cross-functional
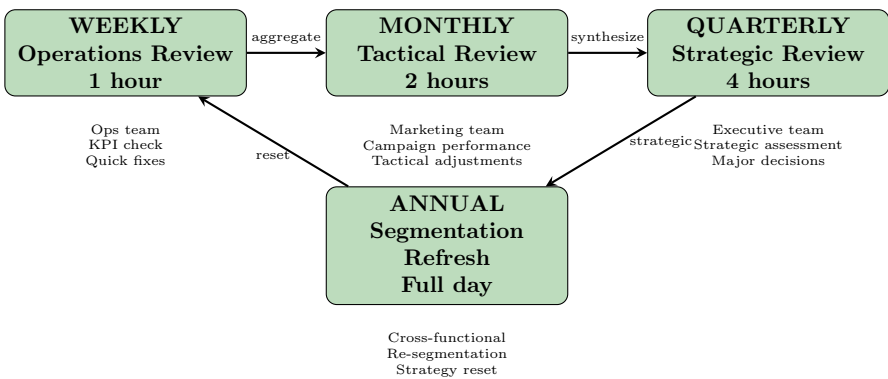Re-segmentation
Strategy reset

Figure 5: Reporting Governance Cycle

## 4.3 Decision Triggers & Thresholds

Table 5: Performance Alert System

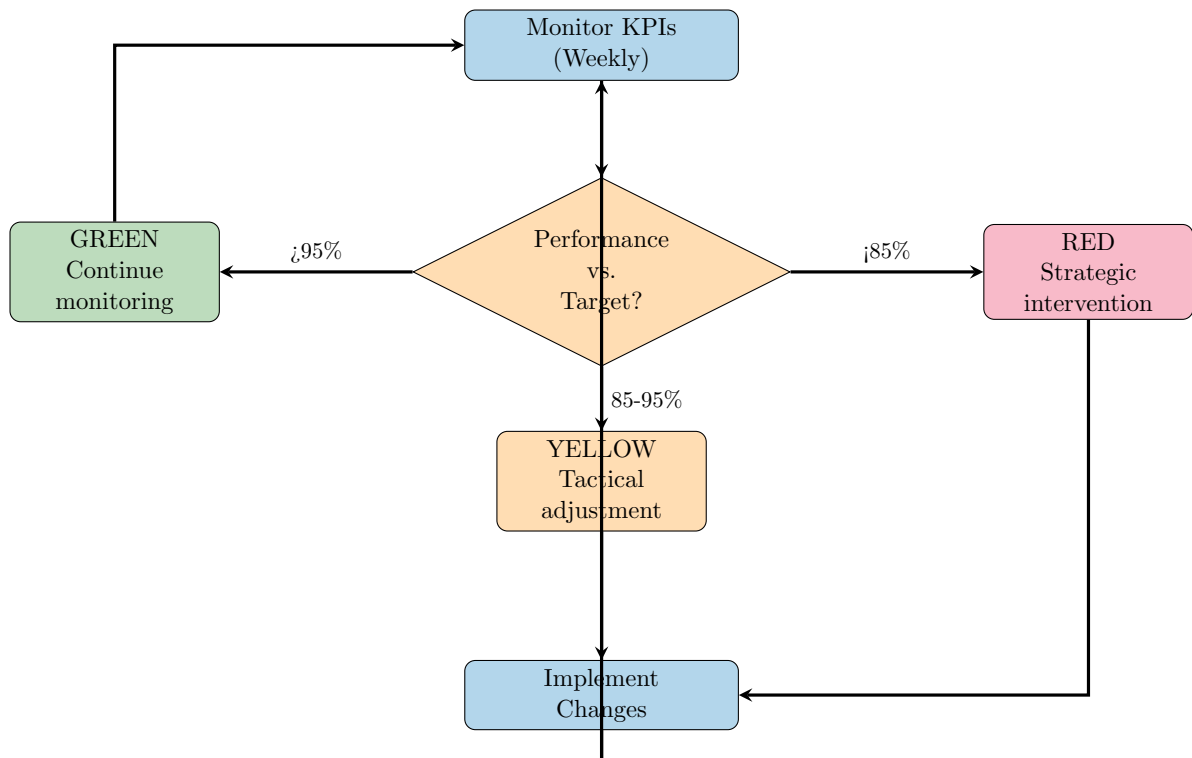| Alert Level | Threshold | Response Time | Action Required |
|---|---|---|---|
| Green | Within 5% of target | Monitor | Continue current strategy |
| Yellow | 5-15% below target | 2 weeks | Tactical adjustment |
| Red | ¿15% below target | Immediate | Strategic intervention |

Figure 6: Performance Alert Decision Flowchart

# 5  Component 4: Continuous Improvement Process
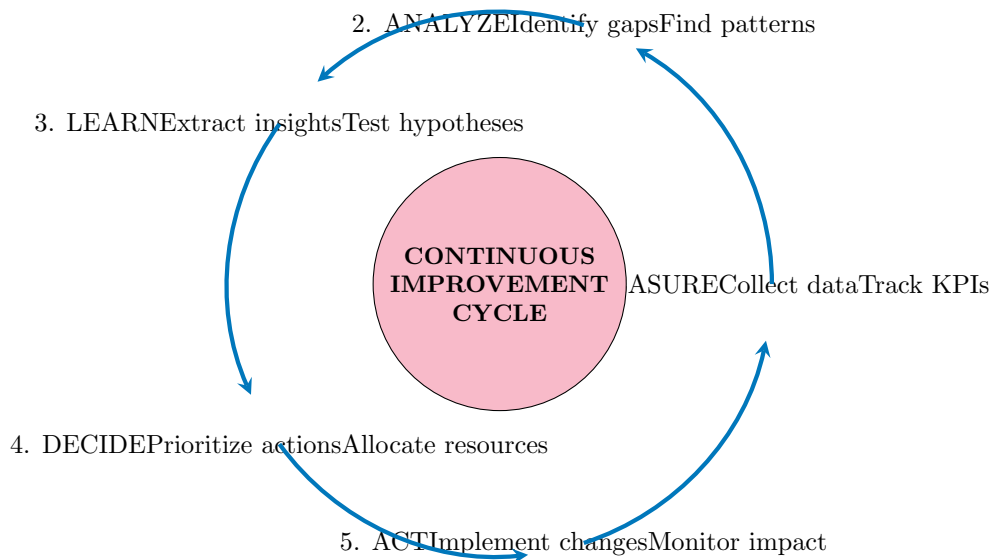
## 5.1  The Optimization Loop



Figure 7: Five-Step Continuous Improvement Cycle

## 5.2    Scenario-Based Optimization Examples

### Scenario 1: Segment 2 Frequency Declining

**Alert:** Visit frequency drops from 2.94 to 2.65 (10% decline)
**Status:** YELLOW alert
**Analysis Steps:**

1. Segment transaction data by time period

2. Identify drop-off patterns (time of day, day of week)

3. Survey sample of segment members

4. Analyze competitive activity during period

**Potential Causes:**

- Competitor value promotions

- Reduced marketing spend

- Service quality issues

- Product availability problems

**Tactical Responses:**

- Launch limited-time value promotion

- Increase digital engagement (app notifications)

- Enhance loyalty program rewards

- Address operational issues if identified

**Timeline:** 2-week implementation, 4-week monitoring

### Scenario 2: Segment 4 Premium Adoption Exceeds Target

**Alert:** 55% trial rate vs. 40% target (38% above)
**Status:** GREEN (positive surprise)
**Analysis Steps:**

1. Understand drivers of over-performance

2. Identify which products most popular

3. Analyze demographic patterns

4. Assess profitability and capacity

**Strategic Opportunities:**

- Expand premium product line faster

- Increase marketing investment in this segment

- Test higher price points

- Extend to adjacent geographies

**Optimization Actions:**

- Accelerate Signature Collection rollout

- Develop additional premium LTOs

- Increase media spend by 20% (reallocate from underperforming)

- Test premium breakfast items

**Timeline:** Quarterly review and expansion planning

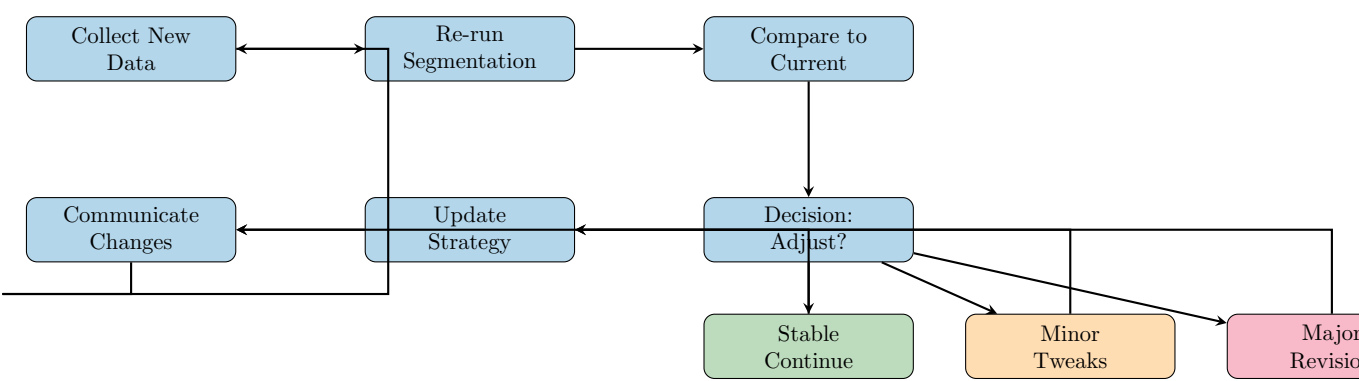## 5.3  Annual Segmentation Refresh Process



Figure 8: Annual Segmentation Refresh Workflow

# 6  Technology & Tool Stack

## 6.1  Recommended Analytics Platform

Table 6: Technology Infrastructure Requirements

| Function | Tool/Platform | Purpose |
|---|---|---|
| Data Warehouse | Google BigQuery / Snowflake | Centralized data storage |
| BI & Dashboards | Tableau / Power BI | Visual analytics & reporting |
| Customer Analytics | Adobe Analytics / Google Analytics | Web & app behavior tracking |
| Survey Platform | Qualtrics / SurveyMonkey | Perception & satisfaction surveys |
| Social Listening | Sprout Social / Brandwatch | Sentiment & trend monitoring |
| Marketing Automation | Salesforce Marketing Cloud | Campaign management & tracking |
| A/B Testing | Optimizely / Google Optimize | Test variations & optimize |
| Data Science | Python/R + Jupyter | Advanced analytics & modeling |

# 7 Organizational Roles & Responsibilities
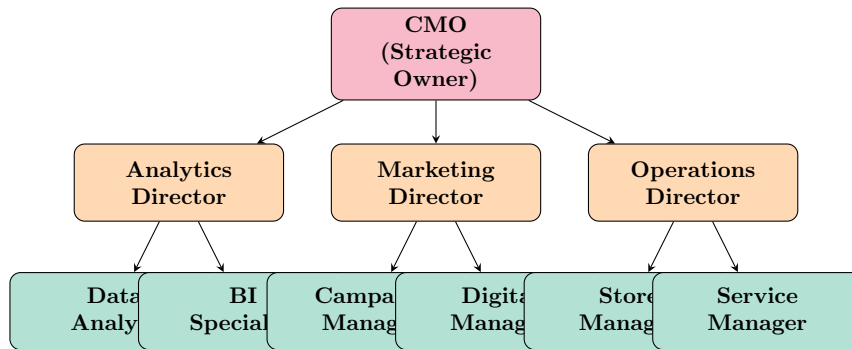


Figure 9: Segmentation Monitoring Organizational Structure

Table 7: RACI Matrix for Monitoring Activities

| Activity | CMO | Analytics | Marketing | Operations |
|---|---|---|---|---|
| Define KPIs | A | R | C | C |
| Collect Data | I | A | C | R |
| Analyze Results | I | R | C | C |
| Generate Reports | I | R | C | C |
| Weekly Review | I | I | R | A |
| Monthly Review | C | C | A | R |
| Quarterly Review | A | R | R | R |
| Strategic Decisions | A | C | C | C |
| Implement Changes | I | C | A | R |

R=Responsible, A=Accountable, C=Consulted, I=Informed

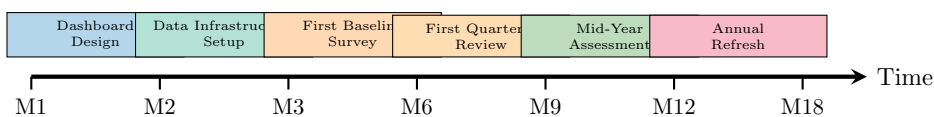# 8 Implementation Timeline: Monitoring Setup



Figure 10: 18-Month Monitoring Implementation Timeline

# 9 Python Implementation: Monitoring System

```python
# Step 10: Automated Performance Monitoring System
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta
import warnings
warnings.filterwarnings('ignore')

```

```
10   print("="*80)
11   print("STEP 10: EVALUATION & MONITORING SYSTEM")
12   print("="*80)
13
14   class SegmentationMonitor:
15       """
16       Comprehensive monitoring system for McDonald's segmentation strategy
17       """
18
19       def __init__(self, target_kpis):
20           self.target_kpis = target_kpis
21           self.alerts = []
22           self.performance_history = []
23
24       def collect_kpis(self, segment_id, period, metrics):
25           """
26           Collect KPI data for specific segment and period
27           """
28           kpi_data = {
29               'segment': segment_id,
30               'period': period,
31               'timestamp': datetime.now(),
32               'metrics': metrics
33           }
34           self.performance_history.append(kpi_data)
35           return kpi_data
36
37       def evaluate_performance(self, current_kpis, segment_id):
38           """
39           Compare current performance against targets
40           """
41           segment_targets = self.target_kpis[segment_id]
42           performance = {}
43
44           for kpi, current_value in current_kpis.items():
45               target_value = segment_targets[kpi]
46               variance = (current_value - target_value) / target_value
47
48               # Determine alert level
49               if variance >= -0.05:  # Within 5% of target
50                   alert_level = 'GREEN'
51               elif variance >= -0.15:  # 5-15% below
52                   alert_level = 'YELLOW'
53               else:  # >15% below
54                   alert_level = 'RED'
55
56               performance[kpi] = {
57                   'current': current_value,
58                   'target': target_value,
59                   'variance_pct': variance * 100,
60                   'alert_level': alert_level
61               }
62
63               # Generate alert if needed
64               if alert_level in ['YELLOW', 'RED']:
```

```python
65                     self.generate_alert(segment_id, kpi, performance[kpi])
66
67             return performance
68
69         def generate_alert(self, segment_id, kpi, details):
70             """
71             Create alert for underperforming KPI
72             """
73             alert = {
74                 'timestamp': datetime.now(),
75                 'segment': segment_id,
76                 'kpi': kpi,
77                 'current': details['current'],
78                 'target': details['target'],
79                 'variance': details['variance_pct'],
80                 'level': details['alert_level'],
81                 'action_required': self.recommend_action(details['alert_level'])
82             }
83             self.alerts.append(alert)
84             return alert
85
86         def recommend_action(self, alert_level):
87             """
88             Recommend action based on alert level
89             """
90             actions = {
91                 'YELLOW': 'Tactical adjustment within 2 weeks',
92                 'RED': 'Strategic intervention - immediate action'
93             }
94             return actions.get(alert_level, 'Continue monitoring')
95
96         def generate_dashboard(self):
97             """
98             Generate performance dashboard visualization
99             """
100            if not self.performance_history:
101                print("No performance data available")
102                return
103
104            # Create dashboard figure
105            fig, axes = plt.subplots(2, 2, figsize=(16, 12))
106            fig.suptitle('McDonald\'s Segmentation Performance Dashboard',
107                         fontsize=16, fontweight='bold')
108
109            # Extract data for visualization
110            segments = [1, 2, 3, 4]
111            segment_names = {
112                1: 'Price-Quality Skeptics',
113                2: 'Happy Value Hunters',
114                3: 'Health-Concerned',
115                4: 'Premium Seekers'
116            }
117
118            # Plot 1: Visit Frequency by Segment
119            ax1 = axes[0, 0]
```

```python
120        # Sample data for visualization
121        visit_freq = [1.40, 2.94, 1.62, 2.86]
122        targets = [2.0, 3.5, 1.8, 3.2]
123
124        x = np.arange(len(segments))
125        width = 0.35
126
127        bars1 = ax1.bar(x - width/2, visit_freq, width, label='Current',
128                        color='steelblue')
129        bars2 = ax1.bar(x + width/2, targets, width, label='Target',
130                        color='lightcoral')
131
132        ax1.set_ylabel('Visit Frequency', fontsize=11)
133        ax1.set_title('Visit Frequency vs Target', fontsize=12,
     ↪  fontweight='bold')
134        ax1.set_xticks(x)
135        ax1.set_xticklabels([f'Seg {s}' for s in segments])
136        ax1.legend()
137        ax1.grid(axis='y', alpha=0.3)
138
139        # Plot 2: Like Ratings by Segment
140        ax2 = axes[0, 1]
141        like_ratings = [1.03, 2.72, 0.23, 2.31]
142        colors = ['orange', 'green', 'red', 'green']
143
144        bars = ax2.bar(segments, like_ratings, color=colors, alpha=0.7,
145                       edgecolor='black', linewidth=2)
146        ax2.axhline(y=0, color='black', linestyle='--', alpha=0.5)
147        ax2.set_ylabel('Like Rating', fontsize=11)
148        ax2.set_title('Brand Affinity by Segment', fontsize=12,
     ↪  fontweight='bold')
149        ax2.set_xlabel('Segment', fontsize=11)
150        ax2.set_xticks(segments)
151        ax2.grid(axis='y', alpha=0.3)
152
153        # Plot 3: Alert Status
154        ax3 = axes[1, 0]
155        alert_counts = {'GREEN': 15, 'YELLOW': 4, 'RED': 1}
156        colors_alert = ['green', 'orange', 'red']
157
158        wedges, texts, autotexts = ax3.pie(alert_counts.values(),
159                                           labels=alert_counts.keys(),
160                                           colors=colors_alert,
161                                           autopct='%1.0f%%',
162                                           startangle=90)
163        ax3.set_title('Current Alert Status', fontsize=12, fontweight='bold')
164
165        # Plot 4: Performance Trend
166        ax4 = axes[1, 1]
167        months = ['M1', 'M2', 'M3', 'M4', 'M5', 'M6']
168        seg2_trend = [2.80, 2.85, 2.90, 2.92, 2.94, 2.96]
169        seg4_trend = [2.70, 2.75, 2.78, 2.82, 2.84, 2.86]
170
171        ax4.plot(months, seg2_trend, marker='o', linewidth=2,
172                 label='Seg 2 (Target)', color='green')
```

```
173        ax4.plot(months, seg4_trend, marker='s', linewidth=2,
174                label='Seg 4 (Target)', color='blue')
175        ax4.axhline(y=3.5, color='green', linestyle='--', alpha=0.5,
176                label='Seg 2 Target')
177        ax4.axhline(y=3.2, color='blue', linestyle='--', alpha=0.5,
178                label='Seg 4 Target')
179        ax4.set_ylabel('Visit Frequency', fontsize=11)
180        ax4.set_title('6-Month Performance Trend', fontsize=12,
       ↪ fontweight='bold')
181        ax4.set_xlabel('Month', fontsize=11)
182        ax4.legend(fontsize=9)
183        ax4.grid(alpha=0.3)
184
185        plt.tight_layout()
186        plt.show()
187
188        print("\n Performance dashboard generated")
189
190    def export_report(self, filename='segmentation_report.csv'):
191        """
192        Export performance data to CSV for stakeholder review
193        """
194        if self.alerts:
195            alerts_df = pd.DataFrame(self.alerts)
196            alerts_df.to_csv(filename, index=False)
197            print(f"\n Alert report exported to {filename}")
198        else:
199            print("\nNo alerts to export")
200
201  # Example usage
202  print("\nInitializing Monitoring System...")
203  print("="*80)
204
205  # Define target KPIs for each segment
206  target_kpis = {
207      1: {'visit_frequency': 2.0, 'like_rating': 1.5, 'transaction_size': 7.0},
208      2: {'visit_frequency': 3.5, 'like_rating': 2.72, 'transaction_size': 7.0},
209      3: {'visit_frequency': 1.8, 'like_rating': 0.5, 'transaction_size': 6.5},
210      4: {'visit_frequency': 3.2, 'like_rating': 2.31, 'transaction_size': 10.0}
211  }
212
213  # Initialize monitor
214  monitor = SegmentationMonitor(target_kpis)
215
216  # Simulate data collection for Segment 2 (Month 3)
217  print("\nCollecting KPI data for Segment 2 (Month 3)...")
218  current_metrics = {
219      'visit_frequency': 2.85,   # Below target of 3.5
220      'like_rating': 2.70,       # Slightly below 2.72
221      'transaction_size': 7.2    # Above target
222  }
223
224  # Evaluate performance
225  performance = monitor.evaluate_performance(current_metrics, segment_id=2)
226
```

```
227  print("\nPerformance Evaluation Results:")
228  print("="*80)
229  for kpi, details in performance.items():
230      print(f"\n{kpi.upper()}:")
231      print(f"  Current: {details['current']:.2f}")
232      print(f"  Target:  {details['target']:.2f}")
233      print(f"  Variance: {details['variance_pct']:+.1f}%")
234      print(f"  Status: {details['alert_level']}")
235
236  # Display alerts
237  if monitor.alerts:
238      print("\n" + "="*80)
239      print("ALERTS GENERATED")
240      print("="*80)
241      for alert in monitor.alerts:
242          print(f"\n[{alert['level']}] Segment {alert['segment']} -
              ↪ {alert['kpi']}")
243          print(f"  Current: {alert['current']:.2f} | Target:
              ↪ {alert['target']:.2f}")
244          print(f"  Variance: {alert['variance']:+.1f}%")
245          print(f"  ACTION: {alert['action_required']}")
246
247  # Generate dashboard
248  print("\n" + "="*80)
249  print("Generating Performance Dashboard...")
250  monitor.generate_dashboard()
251
252  print("\n" + "="*80)
253  print("MONITORING SYSTEM OPERATIONAL")
254  print("="*80)
```

## 10  Key Takeaways from Step 10

> ### Summary of Evaluation & Monitoring Framework
>
> 1. **Four-Component System Established:**
>    - Performance Measurement: 3-level KPI hierarchy
>    - Data Collection: Automated infrastructure
>    - Analysis & Reporting: Dashboard + governance
>    - Continuous Improvement: Closed-loop optimization
>
> 2. **Accountability & Governance:**
>    - Clear roles (RACI matrix)
>    - Regular review cycles (weekly/monthly/quarterly/annual)
>    - Alert system with action thresholds
>    - Executive oversight and strategic decision authority

3. **Technology Enablers:**

   - Integrated data warehouse

   - Real-time dashboards

   - Automated reporting

   - Analytics tools for deep-dive analysis

4. **Continuous Optimization:**

   - Monitor → Analyze → Learn → Decide → Act cycle

   - Scenario-based response playbooks

   - Annual segmentation refresh protocol

   - Adaptive strategy evolution

5. **Complete 10-Step Framework:**

   1. Decided to segment

   2. Specified ideal target

   3. Collected data

   4. Explored data

   5. Extracted segments

   6. Profiled segments

   7. Described with descriptors

   8. Selected targets

   9. Customized marketing mix

   10. Evaluate & monitor

**McDonald's is now equipped with a complete, operational market segmentation strategy supported by systematic monitoring and continuous improvement capabilities.**

## 11 Final Implementation Checklist

Table 8: Step 10 Implementation Checklist

| Task | Status | Notes |
|------|--------|-------|
| Define all KPIs | ☐ | Complete 3-level hierarchy |
| Design dashboard | ☐ | Includes all segments + business metrics |
| Set up data warehouse | ☐ | Integrate POS, app, loyalty, survey data |
| Configure alert system | ☐ | Green/Yellow/Red thresholds |
| Assign roles (RACI) | ☐ | All stakeholders identified |
| Schedule review meetings | ☐ | Weekly/monthly/quarterly calendars |
| Conduct baseline survey | ☐ | Establish T0 measurements |
| Train team on tools | ☐ | Analytics, BI, reporting platforms |
| Document processes | ☐ | SOPs for monitoring activities |
| Launch monitoring system | ☐ | Go-live with full tracking |

## References

[1] Dolnicar, S., Grün, B., and Leisch, F. (2018). *Market Segmentation Analysis: Understanding It, Doing It, and Making It Useful.* Springer.

[2] Kaplan, R.S., and Norton, D.P. (2000). *The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment.* Harvard Business Press.

[3] Davenport, T.H., and Harris, J.G. (2006). *Competing on Analytics: The New Science of Winning.* Harvard Business Press.