

مستندات risk management

نسخه v3.0.0

ارزیابی ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: ارزیابی ریسک

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) | [نسخه beta-2.0.0](#)

این سند به ارزیابی و تحلیل ریسک‌ها می‌پردازد.

تحلیل کیفی

ماتریس احتمال-تأثیر

اولویت	تأثیر	احتمال
بحرانی ●	بالا	بالا
مهم ●	متوسط	بالا

احتمال	تأثیر	اولویت
متوسط	متوسط	متوسط ●

تحلیل کمی

محاسبات مالی

$$\$ \$ EMV = P \times I \times C \$ \$$$

شبیه‌سازی Monte Carlo NEW

```
def monte_carlo_simulation(risks, iterations=10000):  
    results = []  
    for _ in range(iterations):  
        total_cost = sum(  
            risk.probability * risk.impact  
            if random.random() < risk.probability  
            else 0  
            for risk in risks  
        )  
        results.append(total_cost)  
    return results
```

ابزارهای ارزیابی

Microsoft Project •

RISK@ •

NEW Python Risk Analysis •

NEW Machine Learning Models •

[← شناسایی](#) | [مرور کلی](#) | [→ پاسخ به ریسک](#)

شناسایی ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: شناسایی ریسک

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) | [نسخه beta-2.0.0](#)

این سند به روش‌های شناسایی ریسک در پروژه‌ها می‌پردازد.

مقدمه

شناسایی ریسک اولین و مهم‌ترین گام در فرآیند مدیریت ریسک است.

روش‌های شناسایی

1. طوفان فکری (Brainstorming)

جلسات گروهی برای شناسایی ریسک‌های احتمالی.

2. تکنیک دلفی (Delphi)

جمع‌آوری نظرات خبرگان به صورت ناشناس.

3. مصاحبه

مصاحبه با ذینفعان و خبرگان پروژه.

4. تحلیل با هوش مصنوعی NEW

استفاده از الگوریتم‌های یادگیری ماشین برای شناسایی خودکار ریسک‌ها.

```
class AIRiskIdentifier:
    def identify_risks_from_text(self, project_description: str) -> List[str]:
        """شناسایی خودکار ریسک‌ها از متن پروژه"""
        # برای استخراج ریسک‌ها NLP استفاده از
        risks = []
        keywords = ['کمبود', 'مشکل', 'خطر', 'نقص']

        for keyword in keywords:
            if keyword in project_description:
                risks.append(f"ریسک مربوط به {keyword}")

        return risks
```

چک‌لیست شناسایی ریسک

- [] بررسی اسناد مشابه
- [] جلسه طوفان فکری
- [] مصاحبه با خبرگان
- [] تحلیل SWOT
- [] تحلیل با AI NEW

[بازگشت](#) | [مرور کلی](#) | [ارزیابی ریسک](#)

کاهش ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: کاهش ریسک

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) | [نسخه beta-2.0.0](#)

این سند به استراتژی‌های کاهش و مدیریت ریسک می‌پردازد.

مقدمه

کاهش ریسک فرآیندی است که در آن تلاش می‌شود احتمال وقوع یا تأثیر ریسک‌ها کاهش یابد.

روش‌های کاهش

1. کاهش احتمال

تکنیک‌های عملیاتی

- آموزش تیم
- استانداردسازی فرآیندها
- اتوماسیون وظایف
- نظارت مستمر

مثال کد Python

```

class RiskMitigation:
    def __init__(self, risk_name, probability, impact):
        self.risk_name = risk_name
        self.probability = probability
        self.impact = impact
        self.mitigation_actions = []

    def add_action(self, action, cost, effectiveness):
        """افزودن اقدام کاهش‌دهنده"""
        self.mitigation_actions.append({
            'action': action,
            'cost': cost,
            'effectiveness': effectiveness
        })

    def calculate_residual_risk(self):
        """محاسبه ریسک باقیمانده"""
        total_effectiveness = sum(
            action['effectiveness']
            for action in self.mitigation_actions
        )

        residual_probability = self.probability * (1 - total_effectiveness)
        return residual_probability * self.impact





# مثال استفاده
risk = RiskMitigation('تأخیر در تحویل', 0.7, 100000)
risk.add_action('استخدام نیروی اضافی', 20000, 0.4)
risk.add_action('اتوماسیون فرآیند', 50000, 0.3)

print(f"ریسک باقیمانده: ${risk.calculate_residual_risk()}")

```

2. کاهش تأثیر

استراتژی‌های محافظتی

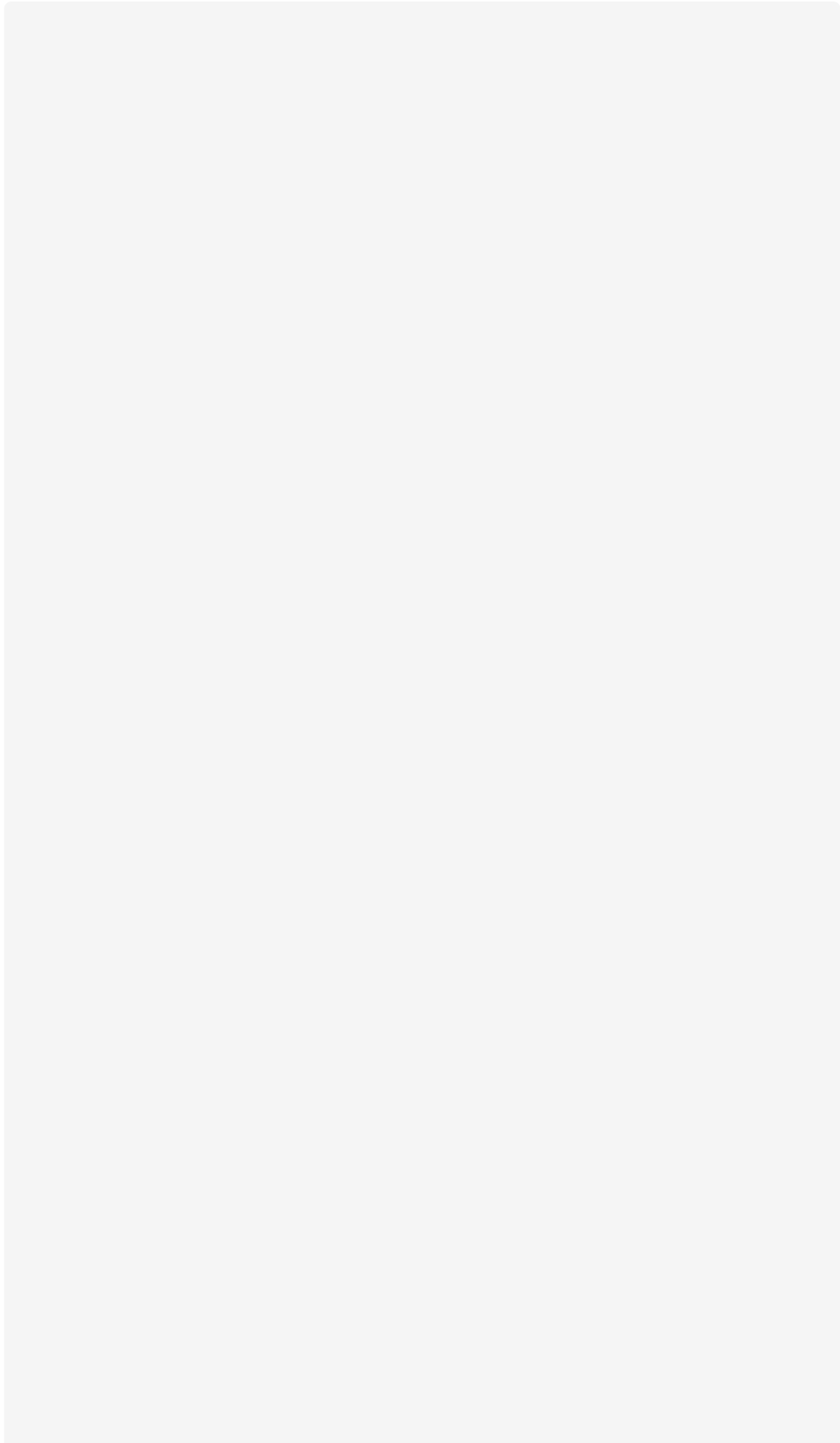
- بیمه و انتقال ریسک 
- پشتیبان‌گیری و بازیابی 
- برنامه‌های اضطراری 
- تنوع‌سازی منابع 

3. برنامه کاهش جامع

مرحله	فعالیت	مسئول	زمان	هزینه
1	شناسایی ریسک‌های کلیدی	مدیر ریسک	هفته 1	\$5,000
2	تحلیل و اولویت‌بندی	تیم تحلیل	هفته 2	\$8,000
3	طراحی اقدامات کاهش	تیم پروژه	هفته 3-4	\$15,000
4	اجرای اقدامات	تیم اجرایی	ماه 2	\$50,000
5	نظارت و ارزیابی	مدیر ریسک	مداوم	\$3,000/ماه

ابزارهای کاهش ریسک

داشبورد تعاملی




```
// نمونه کد برای داشبورد کاهش ریسک
class MitigationDashboard {
  constructor() {
    this.mitigationPlans = [];
  }

  addPlan(riskId, actions, budget) {
    const plan = {
      riskId: riskId,
      actions: actions,
      budget: budget,
      status: 'در حال برنامه‌ریزی',
      effectiveness: 0
    };

    this.mitigationPlans.push(plan);
    return plan;
  }

  updateProgress(planId, newStatus, effectiveness) {
    const plan = this.mitigationPlans.find(p => p.id === planId);
    if (plan) {
      plan.status = newStatus;
      plan.effectiveness = effectiveness;
      this.calculateROI(plan);
    }
  }

  calculateROI(plan) {
    const riskReduction = plan.effectiveness * this.riskSeverityMap[plan.riskId];
    plan.roi = ((riskReduction - plan.budget) / plan.budget) * 100;
    return plan.roi;
  }
}
```

تکنیک‌های پیشرفته NEW

یادگیری ماشین برای پیش‌بینی اثربخشی

```
from sklearn.ensemble import RandomForestRegressor
import numpy as np

class MitigationPredictor:
    def __init__(self):
        self.model = RandomForestRegressor(n_estimators=100)

    def train(self, historical_data):
        """آموزش مدل با داده‌های تاریخی"""
        X = historical_data[['cost', 'complexity', 'timeline']]
        y = historical_data['effectiveness']
        self.model.fit(X, y)

    def predict_effectiveness(self, cost, complexity, timeline):
        """پیش‌بینی اثربخشی اقدام کاهش‌دهنده"""
        features = np.array([[cost, complexity, timeline]])
        return self.model.predict(features)[0]

# مثال
predictor = MitigationPredictor()
# predictor.train(historical_data)
effectiveness = predictor.predict_effectiveness(30000, 5, 12)
print(f"اثربخشی پیش‌بینی شده: {effectiveness:.2%}")
```

معیارهای موفقیت

شاخص‌های کلیدی عملکرد (KPI)

1. نرخ کاهش ریسک:

$$(Initial Risk - Residual Risk) / Initial Risk \times 100\%$$

2. بازگشت سرمایه (ROI):

$$(Risk Reduction - Mitigation Cost) / Mitigation Cost \times 100\%$$

3. زمان پاسخ: میانگین زمان از شناسایی تا اجرای اقدام

4. درصد موفقیت: تعداد اقدامات موفق / کل اقدامات $\times 100\%$

مطالعه موردی

پروژه: پیاده‌سازی سیستم ERP

ریسک اولیه: تأخیر 6 ماهه با هزینه \$500,000

اقدامات کاهش: ☒ استخدام مشاور متخصص: \$80,000 - ☒ آموزش کاربران: \$30,000 - ☒ فازبندی پیاده‌سازی: \$20,000

نتیجه: - تأخیر کاهش به 2 ماه - هزینه اضافی: \$150,000 - صرفه‌جویی خالص: \$220,000 - ROI: 169%

چک‌لیست کاهش ریسک

- [] شناسایی ریسک‌های قابل کاهش
- [] تحلیل هزینه-فایده هر اقدام
- [] اولویت‌بندی بر اساس ROI
- [] تخصیص بودجه و منابع
- [] تعیین مسئولیت‌ها
- [] تنظیم برنامه زمانی
- [] اجرای اقدامات

- [] نظارت بر پیشرفت
- [] ارزیابی اثربخشی
- [] مستندسازی درس‌های آموخته

نتیجه‌گیری

کاهش موثر ریسک نیازمند: - 🎯 هدف‌گذاری دقیق - 💰 تخصیص مناسب بودجه - ⌚ زمان‌بندی صحیح - 👥 مشارکت تیمی - 📊 نظارت مستمر

پایش و کنترل ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: پایش و کنترل

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) | [نسخه beta-2.0.0](#)

این سند به پایش و کنترل مستمر ریسک‌ها می‌پردازد.

شاخص‌های کلیدی (KRI)

داشبورد Real-Time

ریسک‌های بحرانی: 3
ریسک‌های متوسط: 12
ریسک‌های کم: 25
میانگین نمره: 0.42

ابزارهای پایش

داشبورد هوشمند NEW

```
class RealTimeDashboard:
    def __init__(self):
        self.risks = []

    def update_metrics(self):
        return {
            'critical': len([r for r in self.risks if r.score > 0.7]),
            'high': len([r for r in self.risks if 0.4 < r.score <= 0.7]),
            'medium': len([r for r in self.risks if r.score <= 0.4]),
        }

    def generate_alerts(self):
        alerts = []
        for risk in self.risks:
            if risk.score > 0.7:
                alerts.append(f"⚠️ هشدار: {risk.name}")
        return alerts
```

فرکانس بازبینی

- ریسک‌های بحرانی: روزانه
- ریسک‌های مهم: هفتگی
- ریسک‌های متوسط: ماهانه

گزارش‌دهی

قالب گزارش هفتگی

1. خلاصه وضعیت
2. ریسک‌های جدید
3. ریسک‌های بسته شده
4. تغییرات در اولویت‌ها
5. پیش‌بینی‌های AI NEW

[← پاسخ](#) | [📄 مرور کلی](#) | [🏠 بازگشت](#)

مدیریت ریسک در پروژه‌ها - نسخه 3.0.0

نسخه فعلی: 3.0 آخرین نسخه

انتخاب نسخه: نسخه 3.0.0 (فعلی) ▼

نسخه: 3.0 | تاریخ انتشار: آبان ۱۴۰۴ (به‌روزرسانی جدید)

این سند به بررسی پیشرفته مدیریت ریسک در پروژه‌های PMO می‌پردازد.

تغییرات نسخه 3.0.0 NEW

- ✨ اضافه شدن بخش تحلیل کمی ریسک

- اضافه شدن شبیه‌سازی Monte Carlo ✨
- داشبورد مدیریت ریسک ✨
- یکپارچه‌سازی با ابزارهای هوش مصنوعی ✨
- بهبود فرمول‌های محاسباتی 🔄

تعریف ریسک

ریسک یک رویداد یا شرایط نامطمئن است که در صورت وقوع، تأثیر مثبت یا منفی بر اهداف پروژه دارد.

انواع ریسک

1. ریسک‌های فنی

- مشکلات تکنولوژی
- عدم کارایی سیستم‌ها
- نقص‌های طراحی
- ریسک‌های امنیت سایبری NEW

2. ریسک‌های مدیریتی


- ضعف در مدیریت پروژه
- تعارض در تیم
- ارتباطات ناکافی
- عدم چابکی سازمانی NEW

3. ریسک‌های مالی

- افزایش هزینه‌ها
- کاهش بودجه
- نوسانات ارزی

- ریسک‌های توری 

4. ریسک‌های خارجی

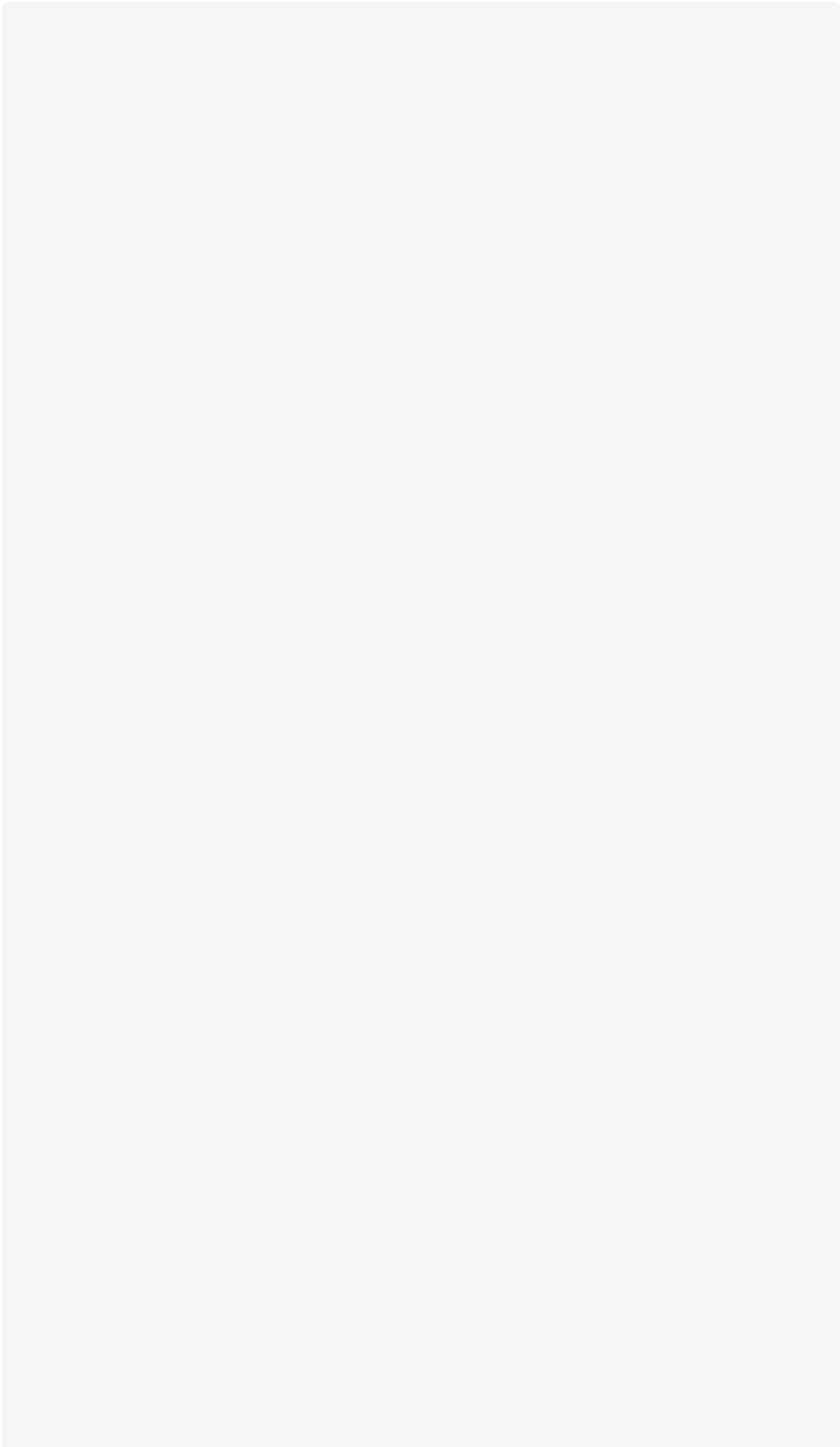
- تغییرات قانونی
- شرایط بازار
- عوامل محیطی
- تحریم‌ها و تغییرات ژئوپلیتیک 

فرآیند مدیریت ریسک پیشرفته

```
graph TD
    A[شناسایی ریسک] --> B[تحلیل کیفی]
    B --> C[تحلیل کمی]
    C --> D[Monte Carlo شبیه‌سازی]
    D --> E[برنامه‌ریزی پاسخ]
    E --> F[AI پایش با]
    F --> A
```

تحلیل کمی ریسک

شبیه‌سازی Monte Carlo



```
import numpy as np
import matplotlib.pyplot as plt

class MonteCarloRiskAnalysis:
    def __init__(self, iterations=10000):
        self.iterations = iterations

    def simulate_cost_risk(self, base_cost, risk_factors):
        """شبيه سازی ريسک هزينه"""
        results = []

        for _ in range(self.iterations):
            total_cost = base_cost
            for risk in risk_factors:
                if np.random.random() < risk['probability']:
                    impact = np.random.uniform(
                        risk['min_impact'],
                        risk['max_impact']
                    )
                    total_cost += impact
            results.append(total_cost)

        return np.array(results)

    def calculate_statistics(self, results):
        """محاسبه آمار نتايج"""
        return {
            'mean': np.mean(results),
            'median': np.median(results),
            'std': np.std(results),
            'percentile_90': np.percentile(results, 90),
            'percentile_95': np.percentile(results, 95)
        }
```

```
# مثال استفاده
risk_factors = [
    {'probability': 0.7, 'min_impact': 10000, 'max_impact': 100000},
    {'probability': 0.5, 'min_impact': 5000, 'max_impact': 50000},
    {'probability': 0.3, 'min_impact': 20000, 'max_impact': 200000}
]

mc = MonteCarloRiskAnalysis()
results = mc.simulate_cost_risk(base_cost=1000000, risk_factors=risk_factors)
stats = mc.calculate_statistics(results)

print(f"میانگین هزینه: {stats['mean']:, .0f}")
print(f"90٪ اطمینان: {stats['percentile_90']:, .0f}")
```

ماتریس احتمال-تأثیر پیشرفته

خیلی زیاد	زیاد	متوسط	کم	خیلی کم	احتمال \ تأثیر
 فاجعه	 بحرانی	 بحرانی	 بالا	 متوسط	خیلی زیاد (<80%)
 بحرانی	 بحرانی	 بالا	 متوسط	 پایین	زیاد (60-80%)
 بحرانی	 بالا	 متوسط	 متوسط	 پایین	متوسط (40-60%)
 بالا	 بالا	 متوسط	 پایین	 پایین	کم (20-40%)

خیلی زیاد	زیاد	متوسط	کم	خیلی کم	احتمال \ تأثیر
بالا 	متوسط 	پایین 	پایین 	پایین 	خیلی کم (%20>)

فرمول‌های پیشرفته

محاسبه ارزش مورد انتظار (EMV):

$$EMV = \sum_{i=1}^n (P_i \times I_i)$$

محاسبه ذخیره احتیاطی (Contingency Reserve):





$$CR = \sqrt{\sum_{i=1}^n (P_i \times I_i^2)}$$

شاخص عملکرد ریسک (RPI):

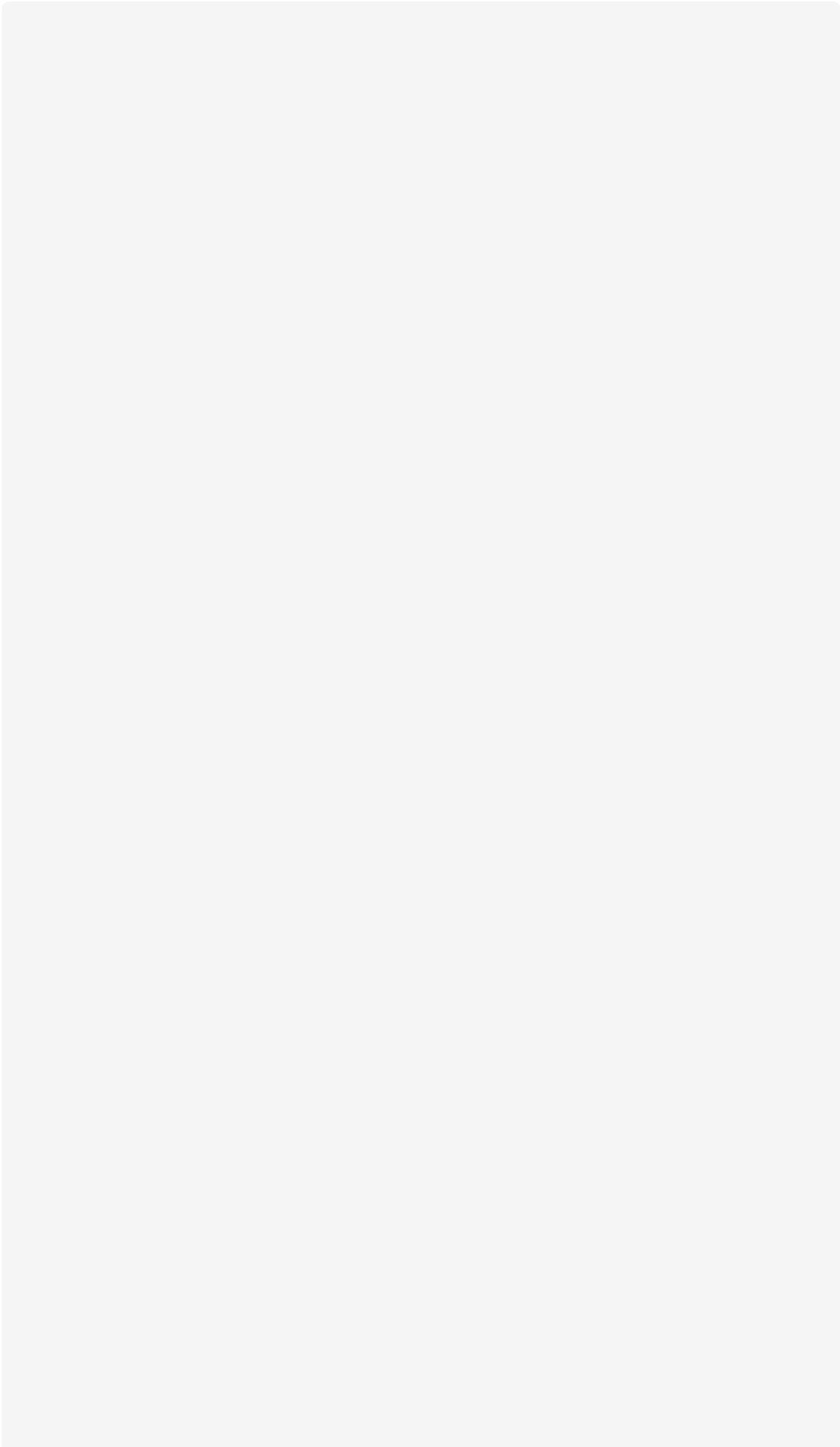
$$RPI = \frac{\text{Risks Closed}}{\text{Total Identified Risks}} \times 100$$

داشبورد مدیریت ریسک

شاخص‌های کلیدی (KRI)

وضعیت	آستانه	مقدار فعلی	شاخص
سبز 	$5 \geq$	3	تعداد ریسک‌های بحرانی
سبز 	$0.5 \geq$	0.42	میانگین نمره ریسک
سبز 	$60\% \leq$	65%	درصد ریسک‌های بسته شده
سبز 	$3 \geq$ روز	2.5 روز	زمان پاسخ به ریسک

یکپارچه سازی با هوش مصنوعی



```
class AIRiskPredictor:
    """پیش‌بینی ریسک با یادگیری ماشین"""

    def __init__(self, model_type='random_forest'):
        self.model_type = model_type
        self.model = None

    def train(self, historical_data):
        """آموزش مدل با داده‌های تاریخی"""
        from sklearn.ensemble import RandomForestClassifier

        X = historical_data[['complexity', 'team_size', 'budget', 'risk_level']]
        y = historical_data['risk_occurred']

        self.model = RandomForestClassifier(n_estimators=100)
        self.model.fit(X, y)

    def predict_risk(self, project_features):
        """پیش‌بینی احتمال وقوع ریسک"""
        probability = self.model.predict_proba([project_features])[0][1]

        return {
            'risk_probability': probability,
            'risk_level': self._classify_risk(probability),
            'recommendation': self._get_recommendation(probability)
        }

    def _classify_risk(self, prob):
        if prob > 0.7:
            return 'بالا'
        elif prob > 0.4:
            return 'متوسط'
        return 'پایین'
```

```
def _get_recommendation(self, prob):  
    if prob > 0.7:  
        return 'اقدام فوری و تخصیص منابع اضافی'  
    elif prob > 0.4:  
        return 'نظارت دقیق و برنامه‌ریزی پاسخ'  
    return 'پایش معمول'
```

مطالعه موردی پیشرفته

پروژه: تحول دیجیتال سازمان

ریسک‌های شناسایی شده:

ID	ریسک	احتمال	تأثیر	EMV	استراتژی
R001	مقاومت در برابر تغییر	0.8	\$500K	\$400K	کاهش
R002	نقص فنی سیستم	0.4	\$1M	\$400K	انتقال
R003	تأخیر در تحویل	0.6	\$300K	\$180K	کاهش
R004	افزایش هزینه	0.5	\$600K	\$300K	پذیرش




ذخیره احتیاطی محاسبه شده: \$687,000

نتایج شبیه‌سازی Monte Carlo: - میانگین هزینه: \$90% - 8.2M اطمینان: \$95% - 9.5M
اطمینان: \$10.1M



اقدامات انجام شده با AI: - پیش‌بینی خودکار ریسک‌های جدید - هشدار زودهنگام برای
ریسک‌های بحرانی - بهینه‌سازی تخصیص منابع

ابزارهای پیشرفته

نرم‌افزارها


- Microsoft Project + Risk Analysis
- Primavera Risk Analysis
- RISK Professional@
- RiskyProject Pro
-  IBM Watson Risk Analytics
-  Oracle Crystal Ball
-  Python Risk Libraries

فریم‌ورک‌ها

- ISO 31000:2018
- PMBOK Guide 7th Edition
- PRINCE2 Risk Management
-  COSO ERM Framework
-  Basel III (برای ریسک‌های مالی)



چک‌لیست جامع مدیریت ریسک

مرحله شناسایی

- [] برگزاری جلسات طوفان فکری
- [] تحلیل SWOT
- [] بررسی دروس آموخته
- [] مصاحبه با خبرگان
- []  تحلیل با هوش مصنوعی

مرحله تحلیل



- [] تحلیل کیفی
- [] تحلیل کمی

- []  شبیه‌سازی Monte Carlo
- []  تحلیل حساسیت
- [] محاسبه EMV

مرحله پاسخ

- [] تعیین استراتژی
- [] تخصیص مسئولیت
- [] محاسبه ذخیره احتیاطی
- []  اتوماسیون پاسخ‌ها

مرحله پایش

- [] بروزرسانی رجیستر ریسک
- [] بررسی KRI ها
- []  داشبورد real-time
- []  گزارش‌دهی خودکار

جمع‌بندی و نوآوری‌ها

نکات کلیدی نسخه 3.0.0

- استفاده از شبیه‌سازی Monte Carlo برای پیش‌بینی دقیق‌تر
- یکپارچه‌سازی با هوش مصنوعی برای هشدار زودهنگام
- داشبورد real-time برای تصمیم‌گیری سریع‌تر
- فرمول‌های پیشرفته‌تر برای محاسبات مالی
- اتوماسیون فرآیندهای روتین

نسخه: 3.0

تاریخ آخرین بروزرسانی: آبان ۱۴۰۴

مسئول سند: دفتر مدیریت پروژه (PMO)

[بازگشت به انتخاب نسخه](#) 🏠 | [نسخه 1.0.0](#) 🟢 | [نسخه beta-2.0.0](#) 🔵

گزارش‌دهی ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: گزارش‌دهی

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) 🟢 | [نسخه beta-2.0.0](#) 🔵

این سند به سیستم‌های گزارش‌دهی و ارتباطات ریسک می‌پردازد.

انواع گزارش‌ها

1. گزارش روزانه

مخاطب: تیم پروژه

محتوا: ریسک‌های فوری و اقدامات امروز

2. گزارش هفتگی

مخاطب: مدیر پروژه

محتوا: خلاصه وضعیت ریسک‌ها و روندها

3. گزارش ماهانه

مخاطب: مدیریت ارشد

محتوا: تحلیل استراتژیک و تصمیم‌گیری

قالب گزارش استاندارد

گزارش ریسک ماهانه

دوره: **مهر 1404**

تهیه کننده: **مدیر ریسک**

خلاصه اجرایی

- تعداد ریسک‌های شناسایی شده: 25
- ریسک‌های بحرانی: 3
- ریسک‌های حل شده: 8
- بودجه مصرفی: 150,000\$

ریسک‌های برتر این ماه

1. **تأخیر در تأمین تجهیزات**

- احتمال: بالا (80%)
- تأثیر: 200,000\$
- وضعیت: در حال پیگیری

اقدامات انجام شده

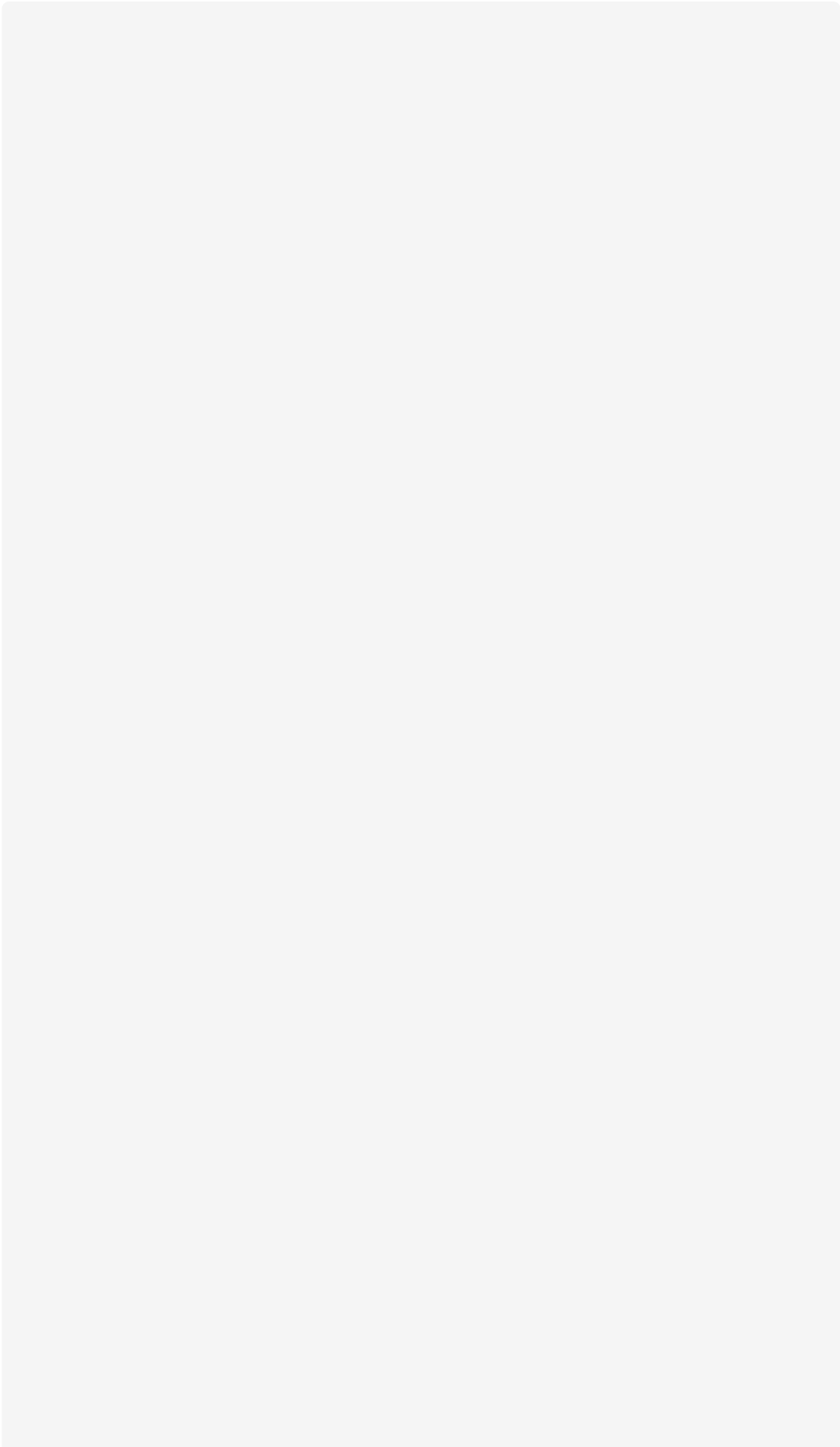
...

توصیه ها

...

داشبورد گزارش دهی

نمودارها و متریک‌ها



```
import matplotlib.pyplot as plt
import pandas as pd

class RiskReportGenerator:
    def __init__(self, risks_df):
        self.risks = risks_df

    def generate_summary_report(self):
        """تولید گزارش خلاصه"""
        summary = {
            'total': len(self.risks),
            'critical': len(self.risks[self.risks['severity'] == 'critical']),
            'high': len(self.risks[self.risks['severity'] == 'high']),
            'medium': len(self.risks[self.risks['severity'] == 'medium']),
            'low': len(self.risks[self.risks['severity'] == 'low'])
        }
        return summary

    def plot_risk_trend(self):
        """رسم روند ریسکها در طول زمان"""
        plt.figure(figsize=(12, 6))

        # Group by date and severity
        trend = self.risks.groupby(['date', 'severity']).size()

        trend.plot(kind='area', stacked=True, alpha=0.7)
        plt.title('روند ریسکها در طول زمان', fontsize=14)
        plt.xlabel('تاریخ')
        plt.ylabel('تعداد ریسک')
        plt.legend(title='شدت', loc='upper left')
        plt.grid(True, alpha=0.3)

        return plt
```

```
def create_risk_matrix(self):
    """ایجاد ماتریس ریسک"""
    matrix = pd.crosstab(
        self.risks['impact'],
        self.risks['probability'],
        margins=True
    )
    return matrix

def export_to_excel(self, filename):
    """خروجی Excel"""
    with pd.ExcelWriter(filename, engine='openpyxl')
        # Summary sheet
        summary_df = pd.DataFrame([self.generate_summary()])
        summary_df.to_excel(writer, sheet_name='خلاصه')

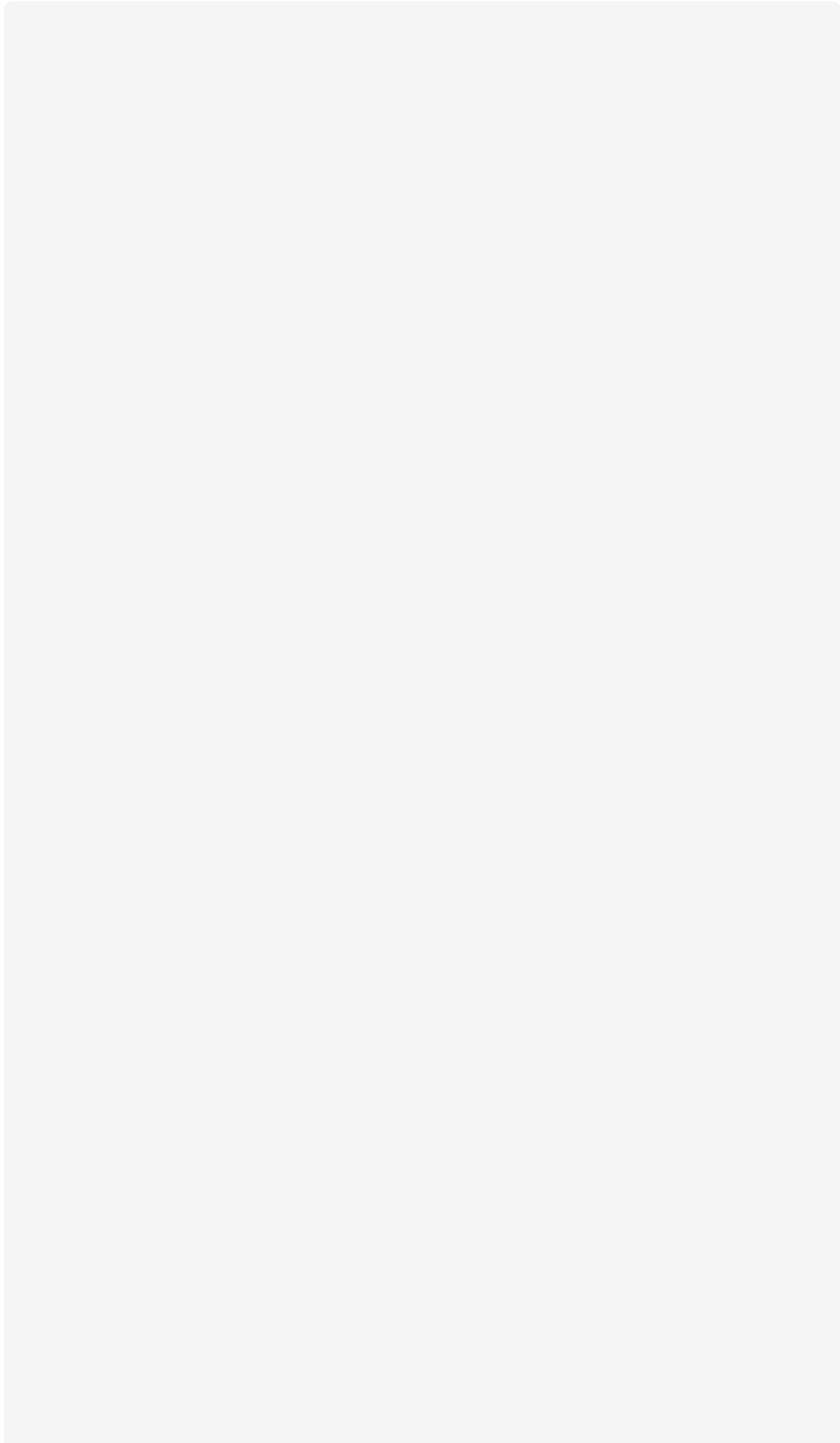
        # Detailed risks
        self.risks.to_excel(writer, sheet_name='ریسک‌ها')

        # Risk matrix
        matrix = self.create_risk_matrix()
        matrix.to_excel(writer, sheet_name='ماتریس')

# استفاده
# risks_df = pd.read_csv('risks.csv')
# reporter = RiskReportGenerator(risks_df)
# reporter.export_to_excel('monthly_report.xlsx')
```

گزارش‌دهی بصری

قالب PowerPoint خودکار



```
from pptx import Presentation
from pptx.util import Inches, Pt

class RiskPresentationGenerator:
    def __init__(self):
        self.prs = Presentation()
        self.prs.slide_width = Inches(10)
        self.prs.slide_height = Inches(7.5)

    def add_title_slide(self, title, subtitle):
        """اضافه کردن اسلاید عنوان"""
        slide_layout = self.prs.slide_layouts[0]
        slide = self.prs.slides.add_slide(slide_layout)

        title_shape = slide.shapes.title
        subtitle_shape = slide.placeholders[1]

        title_shape.text = title
        subtitle_shape.text = subtitle

    def add_risk_summary(self, summary_data):
        """اضافه کردن خلاصه ریسکها"""
        slide_layout = self.prs.slide_layouts[5] # Blank
        slide = self.prs.slides.add_slide(slide_layout)

        # Add title
        title = slide.shapes.add_textbox(
            Inches(0.5), Inches(0.5),
            Inches(9), Inches(0.8)
        )
        title.text = "خلاصه وضعیت ریسکها"
        title.text_frame.paragraphs[0].font.size = Pt(32)
        title.text_frame.paragraphs[0].font.bold = True
```

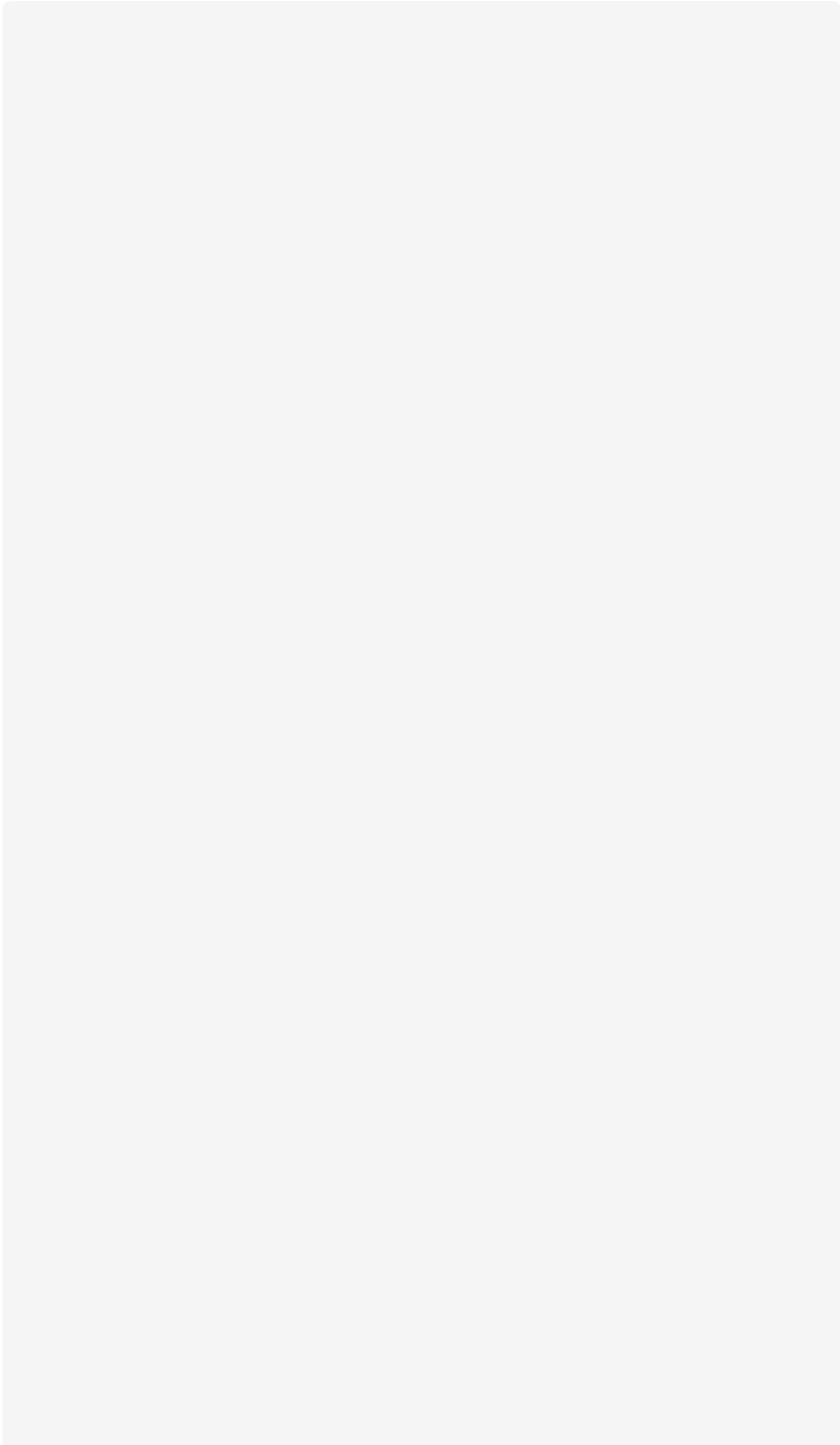
```
# Add summary boxes
y_pos = 1.5
for category, count in summary_data.items():
    self._add_summary_box(slide, category, count,
                          y_pos += 1

def save(self, filename):
    """ذخیره فایل PowerPoint"""
    self.prs.save(filename)

# استفاده
# gen = RiskPresentationGenerator()
# gen.add_title_slide('مهر 1404', 'گزارش ریسک ماهانه')
# gen.add_risk_summary({'15': 'پایین', '12': 'متوسط', '8': 'بالا'})
# gen.save('risk_report.pptx')
```

سیستم هشدار خودکار

ارسال ایمیل هشدار



```

import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

class RiskAlertSystem:
    def __init__(self, smtp_server, sender_email, password):
        self.smtp_server = smtp_server
        self.sender_email = sender_email
        self.password = password

    def send_critical_alert(self, risk_data, recipient_list):
        """ارسال هشدار برای ریسک بحرانی"""
        subject = f"🚨 هشدار: ریسک بحرانی - {risk_data['name']}"

        body = f"""
        یک ریسک بحرانی شناسایی شده است:

        نام: {risk_data['name']}
        احتمال: {risk_data['probability']}%
        تأثیر: ${risk_data['impact']:,}
        وضعیت: {risk_data['status']}

        اقدام فوری مورد نیاز است!

        لطفاً سیستم مدیریت ریسک را بررسی کنید.
        """

        msg = MIMEMultipart()
        msg['From'] = self.sender_email
        msg['Subject'] = subject
        msg.attach(MIMEText(body, 'plain'))

        with smtplib.SMTP(self.smtp_server, 587) as server:
            server.starttls()

```

```

        server.login(self.sender_email, self.password)

        for recipient in recipient_list:
            msg['To'] = recipient
            server.send_message(msg)
            del msg['To']

def send_weekly_digest(self, summary_data, recipient_email):
    """ارسال خلاصه هفتگی"""
    subject = "📊 خلاصه هفتگی ریسک‌ها"

    body = f"""
    خلاصه وضعیت ریسک‌ها در هفته گذشته:

    - کل ریسک‌ها: {summary_data['total']}
    - ریسک‌های بحرانی: {summary_data['critical']}
    - ریسک‌های جدید: {summary_data['new']}
    - ریسک‌های حل شده: {summary_data['resolved']}

    گزارش کامل را در سیستم مشاهده کنید.
    """

    # Implementation similar to above
    # ...

# استفاده
# alert_system = RiskAlertSystem('smtp.gmail.com', 'risk@company.com')
# alert_system.send_critical_alert(risk_data, ['manager@company.com'])

```

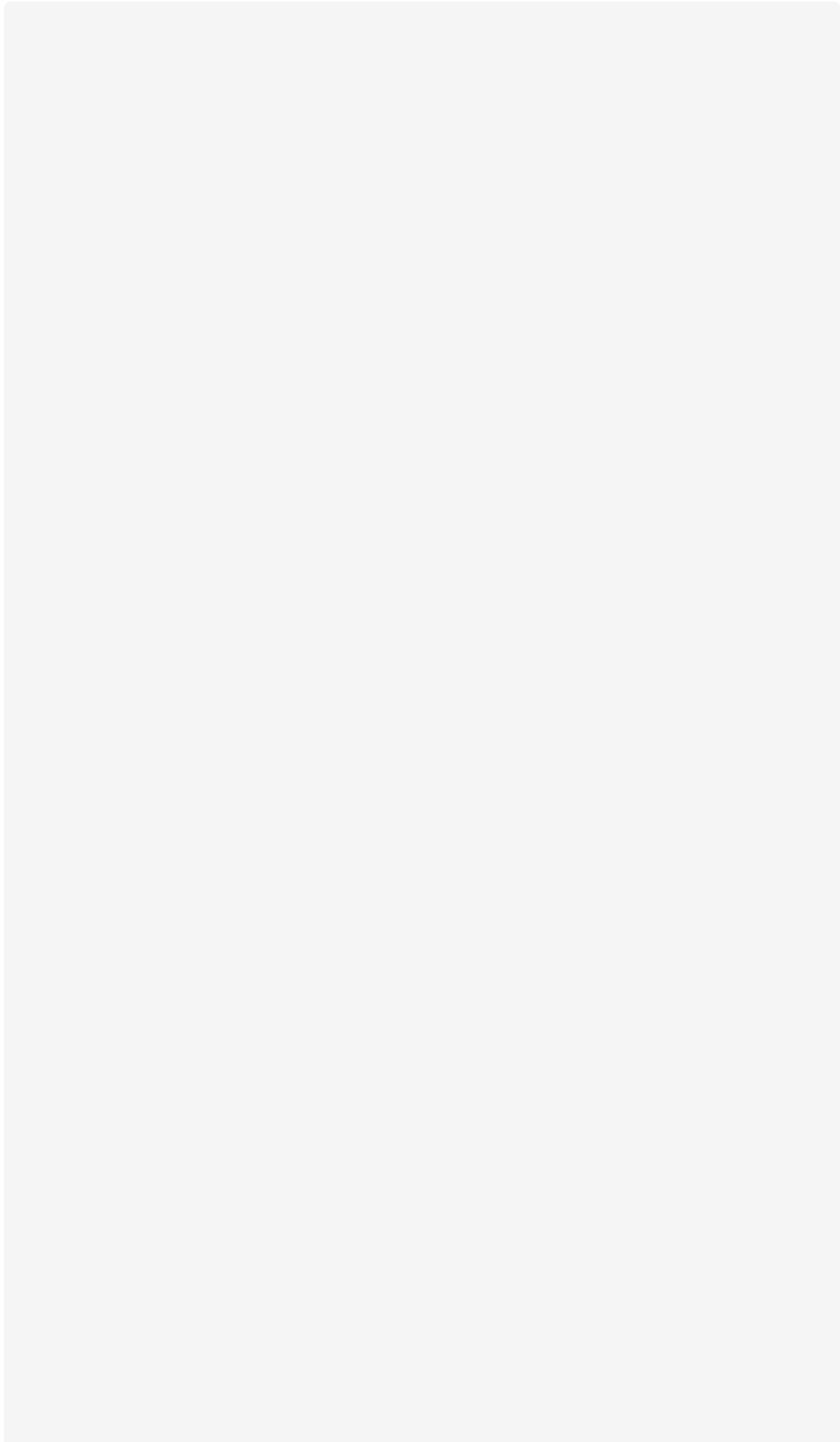
KPI های گزارش‌دهی

معیارهای کلیدی

وضعیت فعلی	هدف	فرمول	معیار
92% ✓	85% <	(ریسک‌های شناسایی قبل از وقوع / کل ریسک‌ها) $\times 100$	نرخ شناسایی به موقع
36 ساعت ✓	48 > ساعت	میانگین زمان از شناسایی تا اقدام	زمان پاسخ میانگین

وضعیت فعلی	هدف	فرمول	معیار
 78%	80% <	(پیش‌بینی‌های صحیح / کل پیش‌بینی‌ها) × 100	دقت پیش‌بینی
 65%	70% <	(ریسک‌های حل شده / کل ریسک‌ها) × 100	نرخ حل ریسک

نمونه Dashboard HTML





```
<!DOCTYPE html>
<html dir="rtl" lang="fa">
<head>
  <meta charset="UTF-8">
  <title>داشبورد ریسک</title>
  <style>
    .dashboard {
      display: grid;
      grid-template-columns: repeat(auto-fit, minma
      gap: 20px;
      padding: 20px;
    }
    .card {
      background: linear-gradient(135deg, #667eea 0
      color: white;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 4px 6px rgba(0,0,0,0.1);
    }
    .card h3 {
      margin: 0 0 10px 0;
      font-size: 1.2em;
    }
    .card .number {
      font-size: 2.5em;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="dashboard">
    <div class="card">
      <h3>کل ریسکها</h3>
      <div class="number">38</div>
```

```
</div>
<div class="card">
  <h3>ریسک بحرانی</h3>
  <div class="number">3</div>
</div>
<div class="card">
  <h3>در حال پیگیری</h3>
  <div class="number">15</div>
</div>
<div class="card">
  <h3>حل شده</h3>
  <div class="number">20</div>
</div>
</div>
</body>
</html>
```

بهترین شیوه‌ها

اصول گزارش‌دهی موثر

1. شفافیت کامل 
2. همه اطلاعات مهم را شامل شود
3. بدون پنهان‌کاری
4. به موقع بودن 
5. گزارش‌ها در زمان مقرر ارسال شود
6. هشدارها فوری باشد
7. قابل فهم بودن 
8. زبان ساده و روان

9. نمودارها و تصاویر گویا

10. قابل اقدام بودن 

11. توصیه‌های مشخص

12. مسئولیت‌های تعیین شده

چک‌لیست گزارش‌دهی

- [] تعیین مخاطبان گزارش
- [] انتخاب قالب مناسب
- [] جمع‌آوری داده‌های دقیق
- [] تحلیل و تفسیر
- [] ایجاد نمودارها
- [] نوشتن خلاصه اجرایی
- [] بررسی و تأیید
- [] ارسال به موقع
- [] دریافت بازخورد
- [] آرشیو و مستندسازی

پاسخ به ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: پاسخ به ریسک

نسخه جاری

شما در حال مشاهده نسخه 3.0.0 (آخرین نسخه) هستید.

نسخه‌های دیگر: [نسخه 1.0.0](#) | [نسخه beta-2.0.0](#)

این سند به استراتژی‌های پاسخ به ریسک می‌پردازد.

استراتژی‌های پاسخ به تهدید

1. اجتناب (Avoid)

حذف کامل ریسک با تغییر برنامه

2. انتقال (Transfer)

انتقال ریسک به طرف سوم (بیمه، پیمانکار)

3. کاهش (Mitigate)

کاهش احتمال یا تأثیر ریسک

4. پذیرش (Accept)

پذیرش ریسک و تخصیص ذخیره احتیاطی

5. اتوماسیون پاسخ NEW

پاسخ خودکار به ریسک‌ها با AI

```
class AutomatedRiskResponse:
    def auto_respond(self, risk):
        if risk.score > 0.8:
            return "اقدام فوری - فعال‌سازی برنامه اضطراری"
        elif risk.score > 0.5:
            return "هشدار - نظارت دقیق"
        else:
            return "پایش معمول"
```

برنامه‌ریزی پاسخ

ریسک	استراتژی	مسئول	بودجه
R001	کاهش	مدیر پروژه	100M
R002	انتقال	مدیر مالی	50M

[← ارزیابی](#) | [📄 مرور کلی](#) | [→ پایش](#)

این مستند به صورت خودکار از سیستم مستندسازی PMO تولید شده است