

مستندات risk management

v3.0.0 نسخه

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: [beta-2.0.0](#) | [نسخه 1.0.0](#)

این سند به ارزیابی و تحلیل ریسک‌ها می‌پردازد.

تحلیل کیفی

ماتریس احتمال-تأثیر

اولویت	تأثیر	احتمال
حرانی	بالا	بالا
مهم	متوسط	بالا

اولویت	تأثیر	احتمال
متوسط 	متوسط	متوسط

تحليل کمی

محاسبات مالی

$$\text{EMV} = P \times I \times C$$

Monte Carlo شبیه‌سازی 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@ •

Python Risk Analysis •

Machine Learning Models •

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

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.

۴. تحلیل با هوش مصنوعی NEW

استفاده از الگوریتم‌های یادگیری ماشین برای شناسایی خودکار ریسک‌ها.

```
class AIRiskIdentifier:

    def identify_risks_from_text(self, project_description):
        """شناസایی خودکار ریسک‌ها از متن پروژه"""
        # برای استخراج ریسک‌ها NLP استفاده از #
        risks = []
        keywords = ['لایحه', '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

[بازگشت](#) | [مرونگلی](#) | [آرزویابی-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:](#) [beta-2.0.0](#) | [نسخه 1.0.0](#)

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 کا هشی"""
        self.mitigation_actions.append({
            'action': action,
            'cost': cost,
            'effectiveness': effectiveness
        })

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

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

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

print(f"ریسک باقیماندہ: ${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	1 هفته	\$5,000
2	تحلیل و اولویت‌بندی	تیم تحلیل	2 هفته	\$8,000
3	طراحی اقدامات کاهشی	تیم پروژه	4-3 هفته	\$15,000
4	اجرای اقدامات	تیم اجرایی	2 ماه	\$50,000
5	ناظارت و ارزیابی	مدیر ریسک	مداوم	\$3,000 /ماه

ابزارهای کاهش ریسک

داشبورد تعاملی


```
// نمونه کد برای داشبورد کاہش 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.getRiskReduction();
        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=10)

    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 کاہشی"""
        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, 10)
print(f"اثربخشی پیش‌بینی شده: {effectiveness:.2%}")
```

معیارهای موفقیت

شاخص‌های کلیدی عملکرد (KPI)

1. نرخ کاهش ریسک:

$$\frac{(\text{Initial Risk} - \text{Residual Risk})}{\text{Initial Risk}} \times 100\%$$

2. بازگشت سرمایه (ROI):

$$\frac{(\text{Risk Reduction} - \text{Mitigation Cost})}{\text{Mitigation Cost}} \times 100\%$$

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

4. درصد موفقیت: تعداد اقدامات موفق / کل اقدامات × 100%

مطالعه موردی

پروژه: پیاده‌سازی سیستم ERP

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

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

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

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

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

نتیجه‌گیری

کاهش موثر ریسک نیازمند: - ⏳ هدف‌گذاری دقیق - 💰 تخصیص مناسب بودجه - ⏳ زمان‌بندی صحیح - 📊 مشارکت تیمی - 📈 ناظارت 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: [beta-2.0.0](#) | [نسخه 1.0.0](#)

این سند به پایش و کنترل مستمر ریسک‌ها می‌پردازد.

شاخص‌های کلیدی (KRI)

Real-Time داشبورد

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
```

فرکанс بازبینی

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

گزارش دهی

قالب گزارش هفتگی

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

[پاسخ](#) | [مرور کلی](#) | [بازگشت](#)   

مدیریت ریسک در پروژه‌ها - نسخه 3.0.0

نسخه فعلی: 3.0 آخرين 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.

3.0.0 تغییرات نسخه

- اضافه شدن بخش تحلیل کمی ریسک

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

تعريف 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
- NEW ریسک‌های امنیت سایبری

2. ریسک‌های مدیریتی

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

3. ریسک‌های مالی

- افزایش هزینه‌ها
- کاهش بودجه
- نوسانات ارزی

ریسک‌های تورمی NEW •

4. ریسک‌های خارجی

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

فرآیند مدیریت ریسک پیشرفته

```
graph TD
    A["[تحلیل کیفی] <--> [شناسایی ریسک]"]
    B["B --> C [تحلیل کمی]"]
    C["C --> D [شبیه‌سازی Monte Carlo]"]
    D["D --> E [برنامه‌ریزی پاسخ]"]
    E["E --> F [پایش با AI]"]
    F["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):
        """ شبیه‌سازی ریسک هزینه """
        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 نتایج """
        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': 1000000},
    {'probability': 0.5, 'min_impact': 5000, 'max_impact': 500000},
    {'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	بحارانی	بحارانی	بالا	متوسط	خیلی زیاد (%80<)
بحارانی	بحارانی	بالا	متوسط	پایین	زیاد (%80-60)
بحارانی	بالا	متوسط	متوسط	پایین	-40 (%60)
بالا	بالا	متوسط	پایین	پایین	کم (%40-20)

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

فرمولهای پیشرفته

محاسبه ارزش مورد انتظار (EMV) :

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

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 ≥	3	تعداد ریسک‌های بحرانی
سبز	0.5 ≥	0.42	میانگین نمره ریسک
سبز	60% ≤	65%	درصد ریسک‌های بسته شده
سبز	3 روز ≥	2.5 روز	زمان پاسخ به ریسک

یکپارچه‌سازی 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', 'size']]
        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 'نظرات دقیق و برنامه ریزی پاسخ'
    return 'پایش معمول'

```

مطالعه موردی پیشرفتہ

پروژه: تحول دیجیتال سازمان

ریسک‌های شناسایی شده:

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

ذخیره احتیاطی محاسبه شده: \$687,000

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

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

ابزارهای پیشرفتہ

نرم افزارها

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 •

(برای ریسک‌های مالی)  Basel III •

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

Monte Carlo NEW [] •

تحلیل حساسیت NEW [] •

EMV [] محاسبه •

مرحله پاسخ

[] تعیین استراتژی •

[] تخصیص مسئولیت •

[] محاسبه ذخیره احتیاطی •

[] اتوماسیون پاسخها NEW [] •

مرحله پایش

[] بروزرسانی رجیستر ریسک •

[] بررسی KRI ها •

real-time داشبورد NEW [] •

[] گزارش‌دهی خودکار NEW [] •

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](#) | [beta-2.0.0](#) | [نسخه 1.0.0](#) | [نسخه 1.0.0](#) | [نحوه استفاده](#)

گزارش دهی ریسک - نسخه 3.0.0

نسخه: 3.0 | بخش: گزارش دهی

نسخه جاری

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

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

این سند به سیستم‌های گزارش دهی و ارتباطات ریسک می‌پردازد.

انواع گزارش‌ها

1. گزارش روزانه

مخاطب: تیم پروژه
محتوی: ریسک‌های فوری و اقدامات امروز

2. گزارش هفتگی

مخاطب: مدیر پروژه
محتوی: خلاصه وضعیت ریسک‌ها و روندها

3. گزارش ماهانه

مخاطب: مدیریت ارشد
محتوی: تحلیل استراتژیک و تصمیم‌گیری

قالب گزارش استاندارد

گزارش ریسک ماهانه #
دوره: * ۱۴۰۴ * مهر
تهیه‌کننده: * مدیر ریسک *

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

ریسک‌های برتر این ماه #
* تأخیر در تأمین تجهیزات * 1.
- احتمال: بالا (%80)
- تأثیر: 200,000\$
- وضعیت: در حال پیگیری -

اقدامات انجام شده

...

توصیه‌ها

...

داشبورد گزارش‌دهی

نمودارها و متريکها


```
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']).sum()

        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='رسکوهای')

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

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

```

گزارش دهی 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('1404', 'مهر ماهانه')
# gen.add_risk_summary({'پایین': 12, 'متوسط': 15, 'بالا': 8})
# 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']}
"""

        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)
            server.sendmail(self.sender_email, recipient_list, msg.as_string())

```

```

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_list):
    """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', 'riskmanagement@example.com')
# alert_system.send_critical_alert(risk_data, ['manager@example.com'])

```

KPI های گزارش دهنده

معیارهای کلیدی

وضعیت فعلی	هدف	فرمول	معیار
<input checked="" type="checkbox"/> 92%	85% <	(ریسک‌های شناسایی قبل از وقوع / کل ریسک‌ها) × 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

Dashboard HTML نمونه


```
<!DOCTYPE html>
<html dir="rtl" lang="fa">
<head>
    <meta charset="UTF-8">
    <title>د اشبورد ریسک</title>
    <style>
        .dashboard {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
            gap: 20px;
            padding: 20px;
        }
        .card {
            background: linear-gradient(135deg, #667eea 0%, #4d5d6e 100%);
            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.

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

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

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

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

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

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