# 流程模型与决策
# Process Models and Decision-Making

Fahad Sabah

2025/12/19

SRE_Fall2025_Beijing University of Technology, China
SRE_Fall2025_北京工业大学，中国

# CONTENTS

流程模型与决策
Process Models and Decision-Making

01 流程框架：敏捷 (Scrum, Kanban), SAFe, ISO/IEC/IEEE 29148.
Process frameworks: Agile (Scrum, Kanban), SAFe, ISO/IEC/IEEE 29148.

02 决策：成本价值分析，AHP。
Decision-making: Cost-value analysis, AHP.

# Why This Matters

Understand why we need structure for teamwork and objective tools for decisions.

**Simple Analogy: Planning a Road Trip**

- Without process: "Let's just drive!" → Chaos, wrong turns, arguments
- Without decision-making: "Where should we eat?" → Endless debate, no one happy
- With both: Map (process) + Restaurant rating app (decision tool) = Successful trip

# 为什么这很重要

理解为什么我们需要团队合作的结构和客观的决策工具。

**简单的比喻：规划一次公路旅行**

- 没有流程：“我们就开车吧！”→混乱、走错路、争吵
- 没有决策：“我们去哪儿吃饭？”→无休止的争论，没人开心
- 两者兼用：地图（流程）+ 餐厅评分应用（决策工具）= 成功出行

# 流程框架——我们的工作方式Process Frameworks – How We Work

## A. Agile (Scrum):

- ❑ The Flexible Mindset
- ❑ **Scrum Example: Building a Mobile App**

Sprint 1 (2 weeks): Basic login screen

Sprint 2 (2 weeks): Add profile picture upload

Sprint 3 (2 weeks): Add friend search feature

**Daily Stand-up: "Yesterday I fixed the login bug. Today I'll work on profile pictures. I'm stuck on image compression."**

## A. 敏捷（Scrum）：

- ❑ 灵活的思维方式
- ❑ Scrum 示例：构建移动应用

冲刺1（2周）：基础登录界面

Sprint 2（两周）：添加头像上传

Sprint 3（2周）：添加好友搜索功能

**每日站谈：“昨天我修复了登录漏洞。今天我要做个人头像。我卡在图像压缩上了。”**

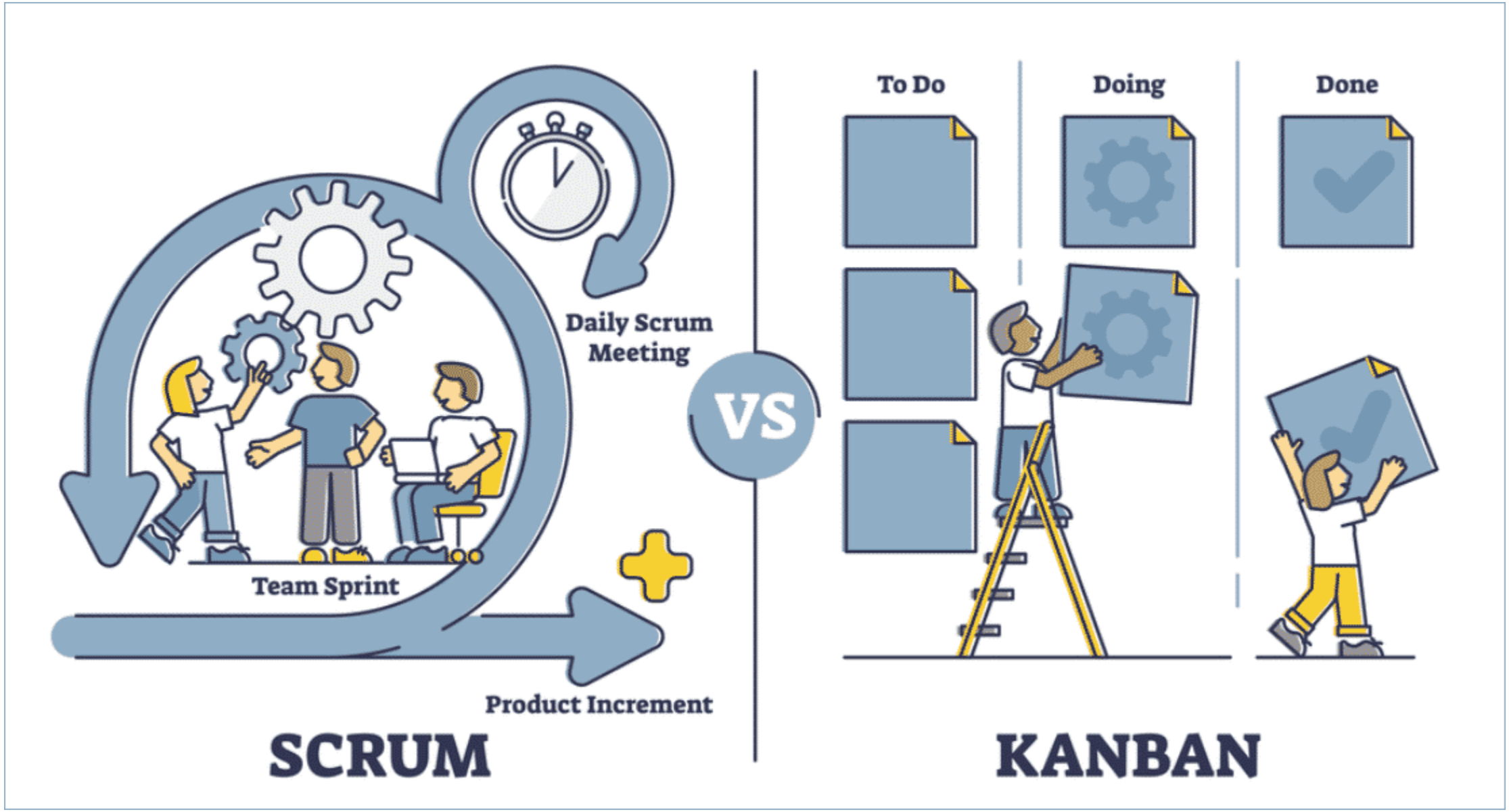
# A. 敏捷（Kanban）：
# A. Agile (Kanban):

- ❑ The Flexible Mindset
- ❑ **Kanban Example:** University IT Help Desk

**Board:**

| TO DO (Max 5) | IN PROGRESS (Max 3) | DONE |
|---|---|---|
| Reset password | Fix printer | Install Excel |
| New email account | Update software | |
| Wi-Fi problem | | |

**Rule: Only 3 tickets in "In Progress" at once → Prevents overload**


Daily Scrum Meeting
Team Sprint
Product Increment
SCRUM
VS
To Do
Doing
Done
KANBAN

# SCRUM

VS

# KANBAN

| | SCRUM | KANBAN |
|---|---|---|
| 1 | Belongs to the Scrum team | Can involve more than one team |
| 2 | No limit on the amount of User Stories | Indicates a maximum number of User Stories to be completed simultaneously |
| 3 | Particularly suitable for one-off projects involving a large team | Particularly suitable for recurring or ongoing projects |
| 4 | Time constraint for completion (Sprint) | No time limit, continuous |
| 5 | Commitment to a number of User Stories to be achieved during the sprint | No commitment, tasks are done one after the other |

# B. SAFe (Scaled Agile Framework):

| | Scrum | SAFe |
|---|---|---|
| **Focus** | Small teams | Organizations |
| **Size** | 7 to 11 members | Multiple teams and teams of teams |
| **Reach** | Doesn't involve middle management | Program and portfolio management |
| **Objective** | Provides structure for small teams | Offers mindset to unify multiple cross-functional teams |
| **Orientation** | Customers | The business |
| **Deliverable** | Optimizes development processes | Improves organizational productivity and reduces inefficiencies |

# C. ISO/IEC/IEEE 29148:

- ❑ The Recipe Book
- ❑ **Example:** Medical Grade vs. Regular Thermometer

**Regular Thermometer:** "Should show temperature"

vs.

**Medical Thermometer (ISO standard):**

1. Must measure 35-42°C ±0.1°C accuracy
2. Must beep when ready
3. Must be waterproof
4. Must display error if battery low
5. [many more precise requirements...]

**Why: Lives depend on it → Need extreme clarity**

# 流程框架一览比较
# Process Frameworks at a Glance Comparison

| Feature | Agile (Scrum, Kanban) | SAFe (Scaled Agile Framework) | ISO/IEC/IEEE 29148 |
|---|---|---|---|
| **Main Focus** | **Team flexibility & customer value.** Deliver small pieces fast, get feedback, and adapt. | **Large-scale alignment.** Coordinate dozens of Agile teams to build a single complex product. | **Requirements clarity & rigor.** Define a formal, consistent process for writing and managing requirements. |
| **Scope** | **Single team or a few teams.** | **Entire organization (Program, Portfolio).** Hundreds of people. | **Any project,** especially in regulated, safety-critical, or complex engineering domains. |
| **Mindset** | **Empirical & Adaptive.** "Let's try, learn, and change." | **Structured & Coordinated.** "Let's align all teams to the business strategy." | **Systematic & Precise.** "Let's define exactly what we need to build, unambiguously." |
| **Key Output** | Working software in short cycles (Sprints). | Integrated solutions from many teams, aligned to business goals. | A formal, validated **Requirements Specification** document. |
| **Best For...** | Startups, small projects, teams needing speed and flexibility. | Large enterprises (e.g., banks, insurers, hardware/software companies) building complex systems. | Regulated industries (medical, aviation, automotive, defense) where safety and compliance are critical. |

# 决策工具——如何选择Decision-Making Tools – How To Choose

**A. Cost-Value Analysis:** Simple Math Choice

**Example 1:** Choosing Weekend Plans with $100 Budget

**Option A: Concert ticket**
- Cost: $80
- Enjoyment value (1-10): 9
- Value per dollar: 9/80 = 0.11

**Option B: Dinner + movie**
- Cost: $60
- Enjoyment value: 7
- Value per dollar: 7/60 = 0.12

**Option C: Game night with friends**
- Cost: $20 (snacks)
- Enjoyment value: 6
- Value per dollar: 6/20 = 0.30

Decision: Game night gives most "fun per dollar"

A. 成本-价值分析：简单的数学选择

**示例1：选择预算为100美元的周末套餐**
**选项A：演唱会门票**
- 成本：80美元
- 乐趣值（1-10）：9
- 每美元价值：9/80 = 0.11

**选项B：晚餐+电影**
- 成本：60美元
- 乐趣值：7
- 每美元价值：7/60 = 0.12

**选项C：和朋友们一起玩游戏之夜**
- 价格：20美元（零食）
- 乐趣值：6
- 每美元价值：6/20 = 0.30

决定：游戏之夜能带来最多“每美元乐趣”

# 决策工具——如何选择Decision-Making Tools – How To Choose

**A. Cost-Value Analysis:** Simple Math Choice

**Example 2:** App Feature Priority (Product Manager View)

**Feature 1: Dark mode**
- Value to users: 8/10
- Development cost (weeks): 2
- Priority score: 8/2 = 4.0

**Feature 2: Voice search**
- Value to users: 9/10
- Development cost: 6 weeks
- Priority score: 9/6 = 1.5

**Feature 3: Share button**
- Value to users: 5/10
- Development cost: 1 week
- Priority score: 5/1 = 5.0

Build order: Share button → Dark mode → Voice search

**A. 成本-价值分析：简单的数学选择**

**示例2：**应用功能优先级（产品经理视图）

**特色一：暗黑模式**
- 对用户的价值：8/10
- 开发成本（周）：2
- 优先级评分：8/2 = 4.0

**功能二：语音搜索**
- 对用户的价值：9/10
- 开发成本：6周
- 优先级评分：9/6 = 1.5

**功能三：分享按钮**
- 对用户的价值：5/10
- 开发成本：1周
- 优先级评分：5/1 = 5.0

构建顺序：分享按钮 → 暗黑模式 → 语音搜索

# 决策工具——如何选择Decision-Making Tools – How To Choose

**B. AHP(Analytic Hierarchy Process):** Choosing When It's Complicated

**Example:** Choosing a Group Project Topic (3 criteria, 3 options)

**Step 1: What matters most? Rate importance:**

1. Easy to complete (1-9 scale vs others)

   vs. Good grade potential: 3 (moderately more important)

   vs. Interesting topic: 1 (equally important)

2. Good grade potential vs. Interesting topic: 5 (much more important)

# 决策工具——如何选择Decision-Making Tools – How To Choose

**Step 2: Compare topics for EACH criteria:**

For "Easy to complete":

Mobile app vs. Website: 2 (slightly easier)

Mobile app vs. Research paper: 5 (much easier)

Website vs. Research paper: 3 (moderately easier)

For "Good grade potential":

Research paper vs. Mobile app: 4 (better grade)

Research paper vs. Website: 6 (much better)

Mobile app vs. Website: 2 (slightly better)

For "Interesting topic":

Mobile app vs. Website: 1 (equally interesting)

Mobile app vs. Research paper: 1/3 (less interesting)

Website vs. Research paper: 1/5 (much less interesting)

# 决策工具——如何选择Decision-Making Tools – How To Choose

**Step 3: Math calculates (software helps):**

Mobile app: 38% score

Website: 25% score

Research paper: 37% score

**Decision: Very close, but mobile app wins slightly**

# 什么是MoSCoW优先级?
# What is MoSCoW Prioritization?

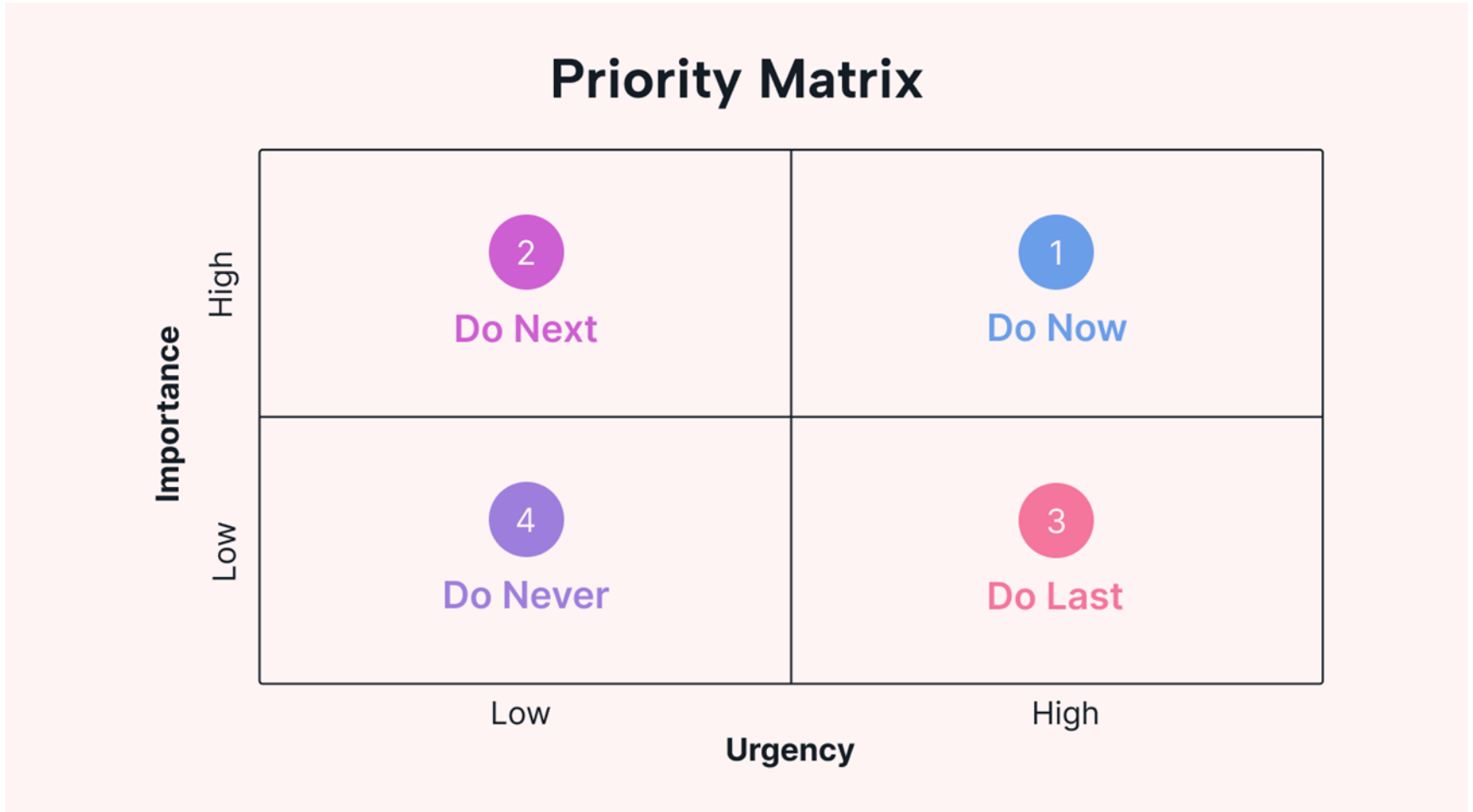
MoSCoW is a prioritization framework (also called MoSCoW analysis) that helps product managers and teams classify features or initiatives into four categories:

Must Have: Features that are critical for the product to function.

Should Have: Features that are important but not essential for immediate implementation.

Could Have: Nice-to-have features that can be added if time allows.

Won't Have: Features that will not be included in this release but may be considered for future iterations.


Priority Matrix
2
Do Next
1
Do Now
4
Do Never
3
Do Last
High
Low
Importance
Low
High
Urgency

# 什么是卡诺模型?
# What is the Kano Model?

The Kano model is a user research technique developed in the 1980s by Professor Noriaki Kano, which categorizes product features into five categories:

1. **Must-have** - or "basic expectations"
2. **Performance** - customer satisfaction increases in line with the quality of the feature
3. **Delighters** - if present, people are surprised and delighted
4. **Reverse** - actively disliked by customers
5. **Indifferent** - nobody cares if these features exist or not

# 主要要点
# Key Takeaways

**When to use what:**

CHAOTIC, CHANGING PROJECT → Agile/Scrum

STEADY FLOW OF TASKS → Kanban

BIG COMPANY, MANY TEAMS → SAFe

LIFE-CRITICAL SYSTEMS → ISO Standards

SIMPLE $$$ DECISIONS → Cost-Value Analysis

COMPLEX MULTI-FACTOR DECISIONS → AHP

**何时使用什么：**

混乱且不断变化的项目→敏捷/Scrum。
看板→任务的稳定流
大公司，很多团队→SAFe
生命关键系统→ISO标准

简单的$$$决策→成本价值分析
复杂的多因素决策→AHP

# 案例研究：MediSecure - 认证与监控子系统
# Case Study: MediSecure - Authentication & Monitoring Subsystem

**1. The Challenge**

MediSecure must enhance security to comply with new regulations and protect sensitive patient data. The initial 90+ stakeholder requirements range from "multi-factor authentication" to "AI-powered anomaly detection." The team has 6 months and a fixed budget. They need a clear, defensible prioritization plan.

**1. 挑战**

MediSecure必须加强安全，以符合新法规并保护敏感患者数据。最初的90+利益相关者需求涵盖了“多因素认证”到“人工智能驱动的异常检测”。团队有6个月时间和固定预算。他们需要一个清晰且可辩护的优先级计划。

# 案例研究：MediSecure - 认证与监控子系统
# Case Study: MediSecure - Authentication & Monitoring Subsystem

2. Applying a Multi-Method Prioritization Framework
A single method isn't enough. The team uses a layered approach: MoSCoW for release planning, Kano for user satisfaction insight, and AHP for complex trade-offs between security attributes.

Phase 1: High-Level Categorization with MoSCoW
The team first classifies requirements into Must, Should, Could, Won't categories based on legal, safety, and business survival criteria.

2. 应用多方法优先级框架
单一方法是不够的。团队采用分层方法：MoSCoW负责发布规划，Kano负责用户满意度洞察，AHP负责安全属性间的复杂权衡。
第一阶段：与MoSCoW的高级分类
团队首先根据法律、安全和商业生存标准，将需求分为“必须”、“应该”、“可以”、“不愿意”等类别。

# 案例研究：MediSecure - 认证与监控子系统
# Case Study: MediSecure - Authentication & Monitoring Subsystem

| MoSCoW Category | Criteria (For this Cybersecurity Context) | Example Requirements for MediSecure |
|---|---|---|
| **MUST HAVE** | Non-negotiable. Required by law (HIPAA), or prevents immediate, critical breaches. | • Role-Based Access Control (RBAC).<br>• Audit logging for all data access.<br>• Encryption of data at rest. |
| **SHOULD HAVE** | High importance for security posture but not legally mandated for launch. Lack would create significant risk. | • Multi-Factor Authentication (MFA) for all external users.<br>• Automated nightly vulnerability scans. |
| **COULD HAVE** | Valuable enhancements that improve efficiency or address moderate risks. | • Biometric authentication option.<br>• A dashboard for real-time threat visualization. |
| **WON'T HAVE** | Recognized as valuable but out of scope due to time/budget, or low ROI for current threat landscape. | • Fully integrated deception technology (honeypots).<br>• Custom AI model for zero-day threat prediction. |

## 案例研究：MediSecure - 认证与监控子系统
## Case Study: MediSecure - Authentication & Monitoring Subsystem

Result: This creates the baseline for Release 1.0: All Must Haves are non-negotiable. Should Haves become the primary target if resources allow.

Phase 2: Understanding User Perception with Kano Model
The team surveys two key user groups: Clinical Staff (Doctors/Nurses) and IT Security Analysts. They analyze features not classified as "Must Have."