S-SDL企业应用实践

关于漏洞、风险、成本

1. 为什么软件总会有漏洞？漏洞是怎么引入的？
   - 业务需求引入
   - 产品设计引入
   - 编码引入
2. 如何应对风险
   - 缓解风险
   - 转移风险
   - 接受风险
3. 安全成本
   - 对于企业来说，安全投入多少是合适的？

什么是S-SDL

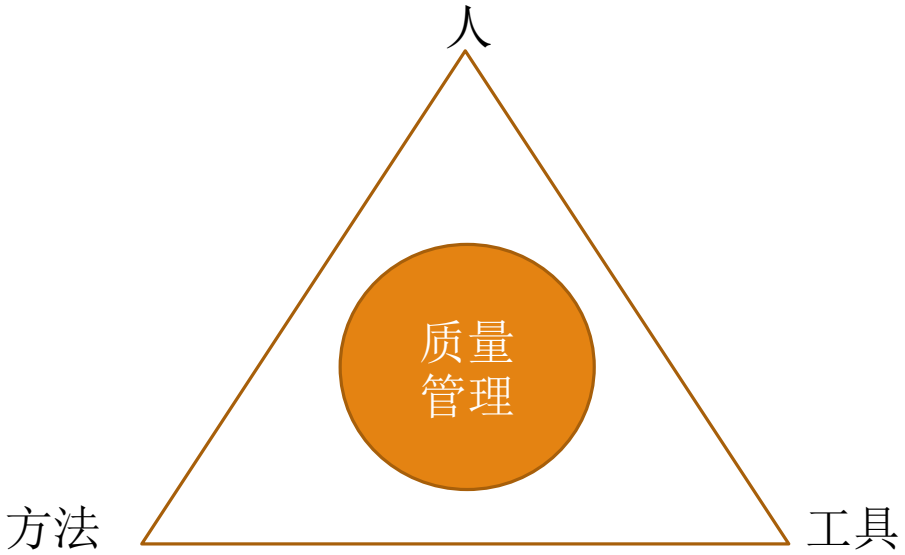
不是一种安全技术，而是E2E的安全工程能力

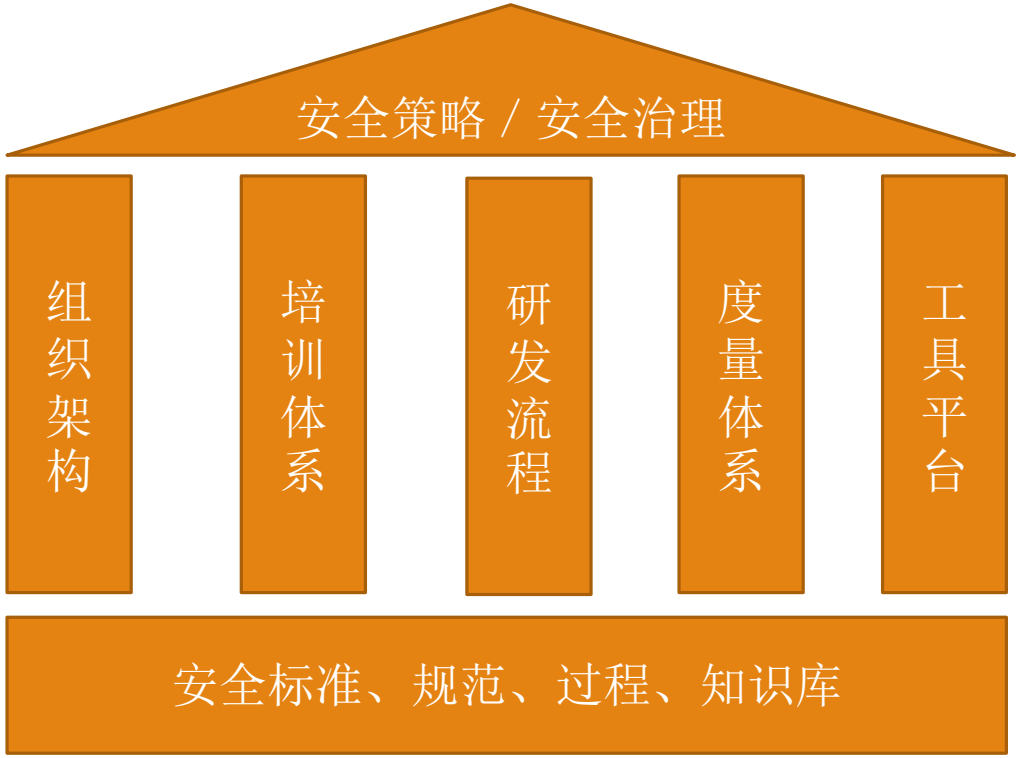
也是一种Security Built In的解决方案

目标：交付更安全的系统

S-SDL架构

安全是质量属性的一部分，将安全融入到质量管理是构建S-SDL的基本条件





软件安全研发流程设计

安全融入开发流程

建立安全需求

1. 分析业务需求对安全的影响
2.来自客户的显性安全需求
3.安全需求基线
3.合规、认证需求

安全设计—Security by Design

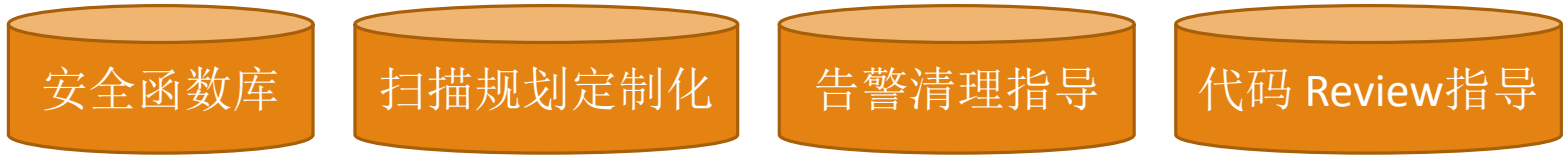
1.攻击面分析
- 攻击面最小化

2. 威胁建模
- STRIDE威胁建模
- 攻击树威胁建模

安全设计原则	说明
开放设计	安全不依赖于设计的秘密
失败安全	基于允许的访问决策
权限分离	一种保护机制需要两把钥匙来解锁
最小授权	根据业务需求最授权
经济适用	越复杂的东西越容易出漏洞
最小公共化	尽量减多用户间公用的且被所有用户依赖的机制
完全仲裁	每一次访问都应该进行权限检查
心理可承受	安全机制的设计要易于使用
不轻信	默认不可信
保护薄弱环节	攻击往往在薄弱点发生
纵深防御	不依赖单一的安全机制

代码安全

1. 安全编码规范
2. 代码扫描及告警清理
3. 代码Review



安全测试

1.基于威胁建模的测试

2.Fuzzing

3.已知漏洞扫描

4. 测试问题跟踪

发布及漏洞管理

1. 安全生态建设—漏洞收集
2. 漏洞分析，排查，预警
3. 根因分析