2020北京网络安全大会
2020 BEIJING CYBER SECURITY CONFERENCE

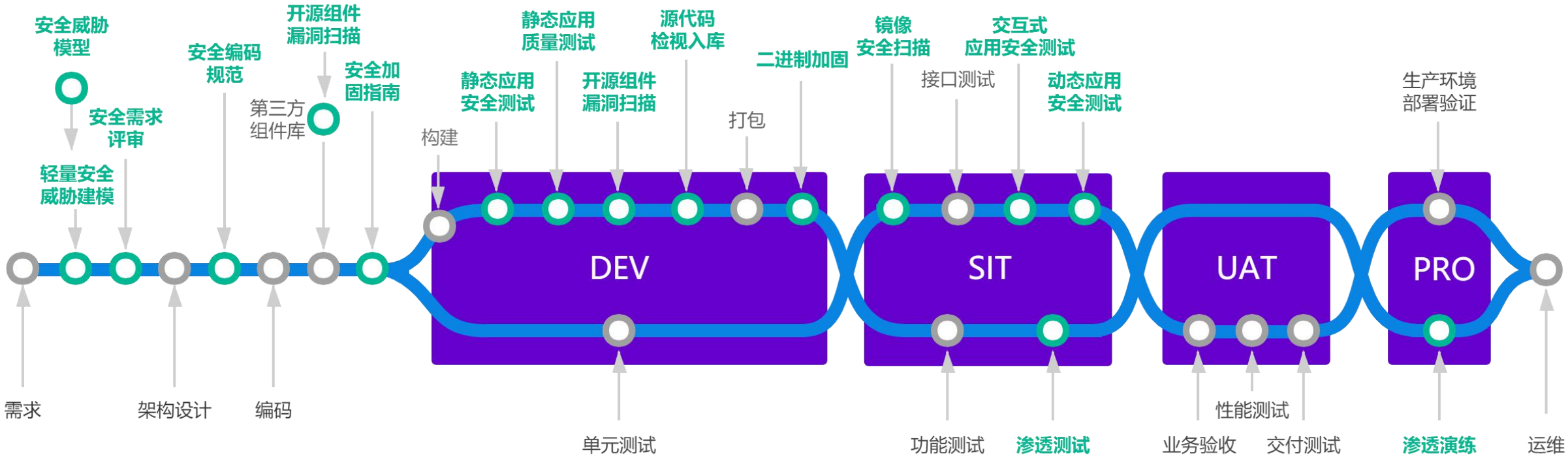
敏捷开发中的开源安全治理

开源安全治理的探索与实践

如何做

开源安全治理的思路

开源安全治理的价值



- DevOps成熟度依赖低
- 较高的威胁确定性
- 足够的威胁震撼性

- 误报低、效率高、易自动化
- 投入产出比可观

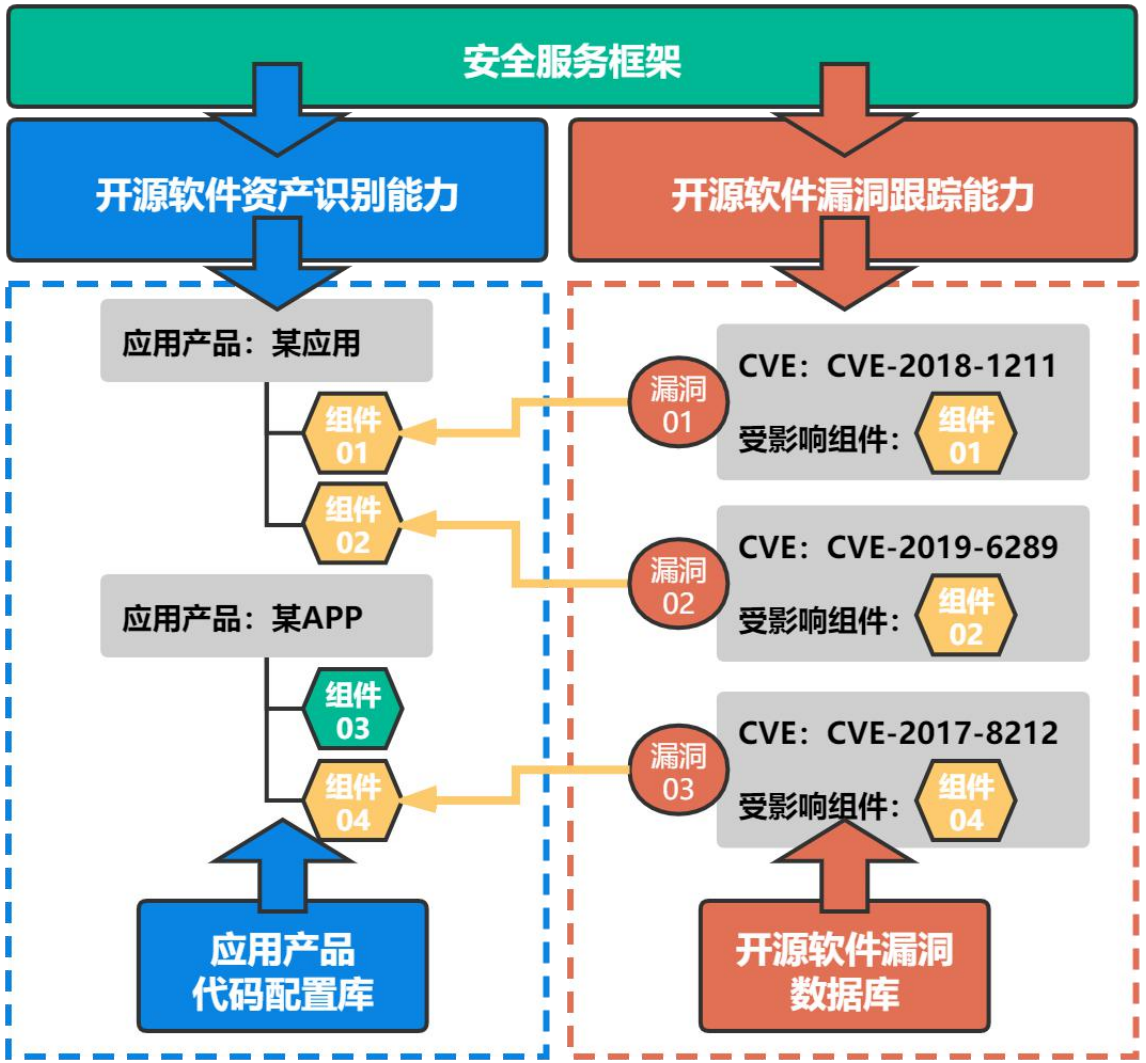
开源安全治理所需能力

研发资产中开源软件及漏洞的识别能力

- **开源识别：**对于给定的配置库、制品库或代码包，能够通过技术手段，快速、准确的识别其中的开源软件及其版本
- **漏洞识别：**对于识别出的开源软件，能够关联并分析该开源软件存在的漏洞信息
- **开源认证：**提供资产软件与官方软件一致性认证能力，避免软件被篡改，确保资产及工具链的安全性

开源软件资产登记及漏洞跟踪能力

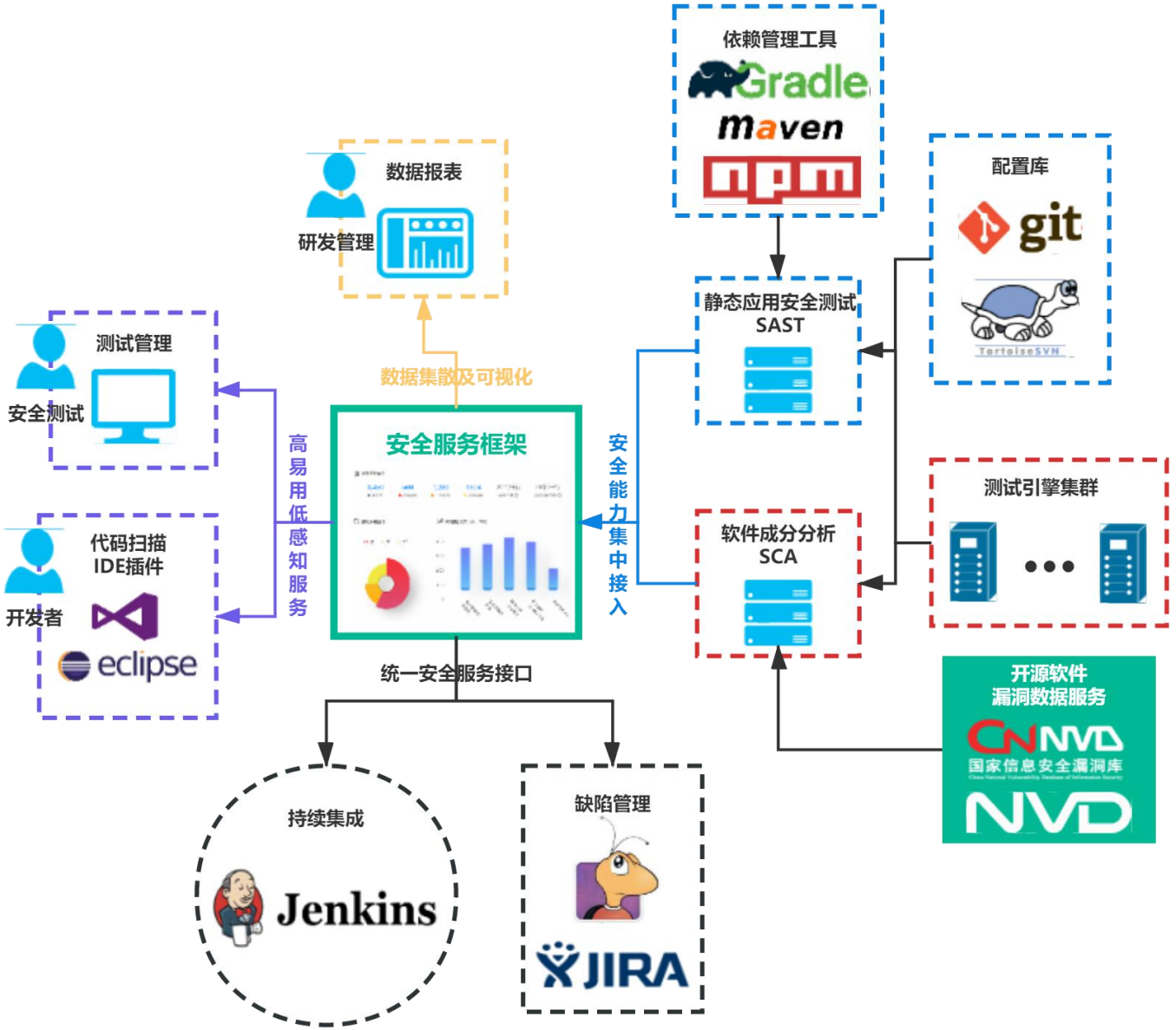
- **资产登记：**根据开源软件识别结果，保存产品及其版本与开源软件间的映射关系
- **漏洞跟踪：**当开源软件漏洞数据更新后，能推送是否有新的漏洞影响系统中正在使用的开源软件，一旦有新的漏洞影响，能及时进行预警，确保不错过漏洞情报



开源安全治理的实现方式

- 安全能力集中接入
- 高易用、低感知的安全服务
- 统一的安全服务接口
- 数据集散及信息可视化

安全服务框架，核心是对多种安全验证工具、能力进行统一封装，并通过API或UI交互，及融入交付流水线等途径，向开发团队提供便捷易用的安全服务集合。通过安全服务框架提供的服务集合，贯穿软件开发的完整生命周期，从需求分析和技术设计，直到上线运营及运维。同时，安全服务框架也服务于安全团队，向其提供一系列安全管理服务。



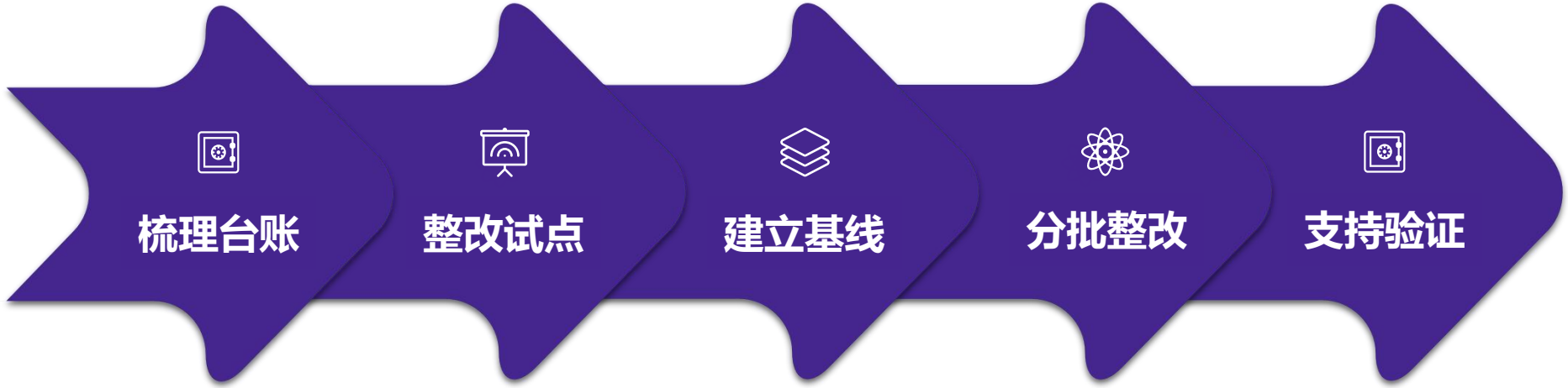
实践与问题

解决历史、管控当下、防患未来

开源安全治理的实践

- 解决历史：存量开源漏洞梳理与整改
- 管控当下：开发流程的开源安全检查
- 防患未来：开源漏洞预警及应急响应

存量开源漏洞梳理与整改



- 检测应用产品生产分支
- 建立开源软件漏洞与产品映射关系
- 梳理数据供管理层整改决策

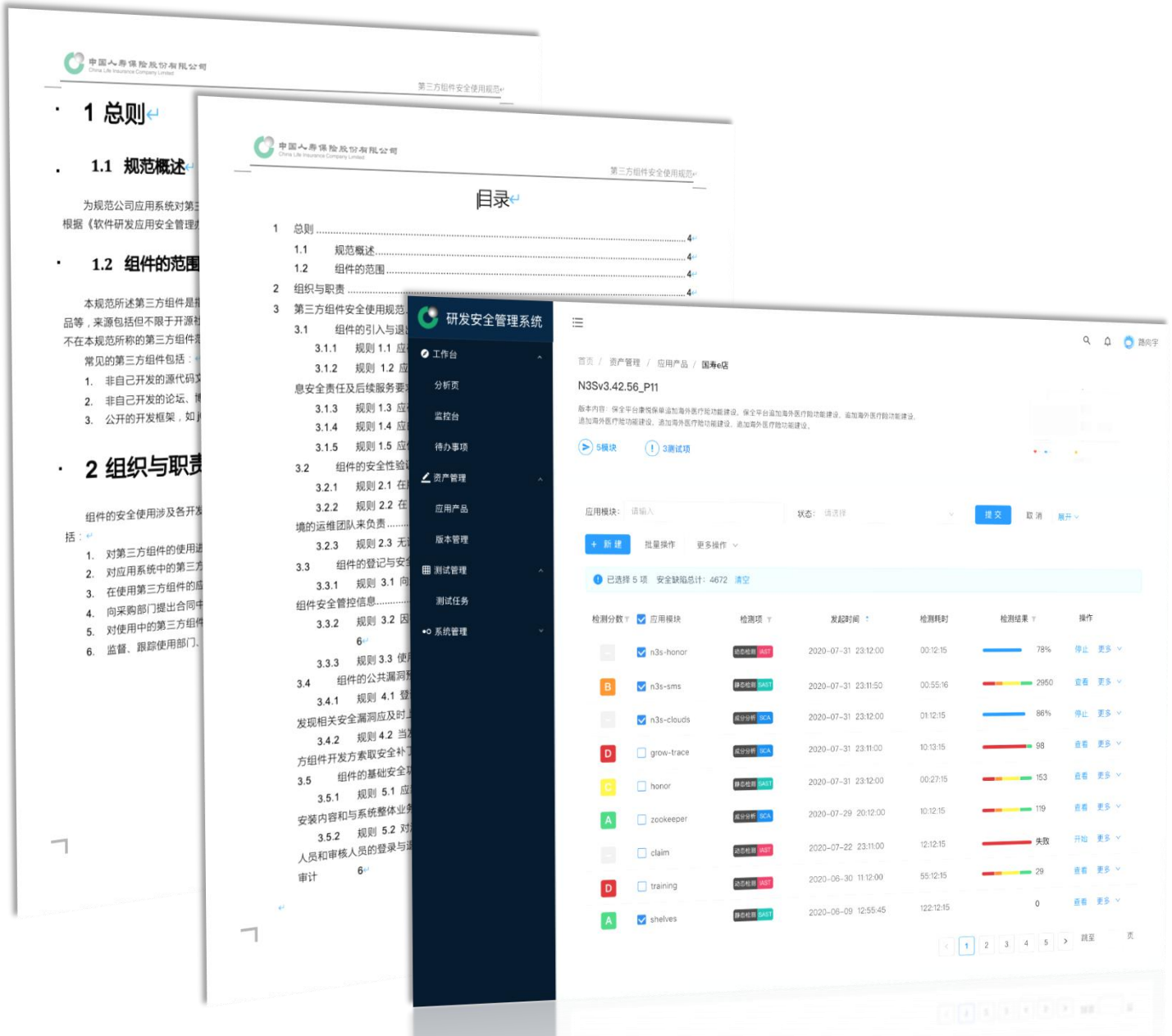
- 框架类
- 依赖JDK升级
- 功能单一类
- 技术可行、风险可控

- 区分存量与增量
- 区分各整改批次

- 漏洞级别和应用形态分批整改，降低风险
- 共性解决方案
- 制定例外流程

- 修复验证
- 漏洞跟踪
- 私有版本的维护
- 推动供应商整改

开发流程的开源安全检查



开箱即用的开源漏洞检测服务

- 通过测试服务框架界面无感知的引入开源漏洞检测项
- 通过统一安全服务接口无缝接入持续集成流程，提供持续的漏洞自检测能力

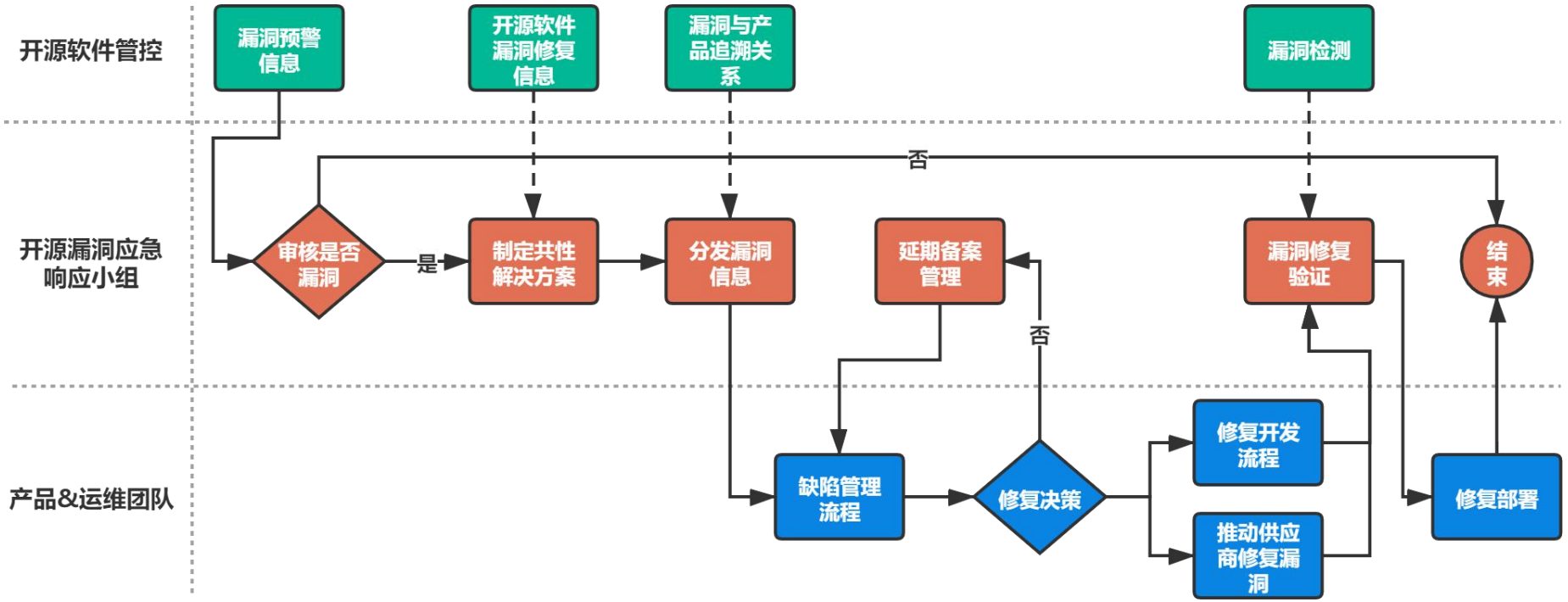
开源软件安全使用规范

- 从引入与退出、使用与配置、安全性验证等多方面规范开源软件的使用

开源漏洞预警及应急响应

漏洞信息	
漏洞名称	Dom4j 代码问题漏洞
组件名称	Dom4j: flexible XML framework for Java
缺陷编号	CLIC-SEC-2020-0506
CVE编号	CVE-2020-10683
漏洞等级	超危
漏洞描述	dom4j是一款用于处理XML的开源框架。 dom4j 2.1.3之前版本中存在代码问题漏洞。该漏洞源于网络系统或产品的代码开发过程中存在设计或实现不当的问题。
类型	资料不足
POC	无
应对策略	
优选策略	组件升级到优选版本
可选策略	无
禁用版本	1.0.0-2.1.2
优选版本	2.1.3
可选版本	无
整改信息	
整改要求	1、优先升级到优选版本； 2、如果需要备案，需明确计划整改日期。
QC缺陷等级	L3（20自然日完成整改）
QC录入方式	安全测试团队录入
缺陷发布日期	2020年6月4日
整改截止日期	2020年6月24日

漏洞预警通知



使用登记　漏洞预警　响应整改

总结

开源安全治理的关键点

总结

开源软件治理投入明确、效果具体，整改风险可控

- 大幅提升产品的安全和质量
- 提升产品的整体技术水平

检测能力是基本，建立资产安全管理更重要

- 快速提供各层面所需的数据
- 组件的使用不是一刀切，不是安全部门说不安全就不能用的
- 方便把控风险管控、风险收敛

贴合实际情况的检测平台或者框架

- 提升测试能力的接入效率
- 降低测试成本
- 测试结果快速确认为产品缺陷

培养产品团队的安全意识

- 开源软件的责任主体是产品团队
- 事后补救永远不如事前预防


2020北京网络安全大会
2020 BEIJING CYBER SECURITY CONFERENCE
THANKS
全球网络安全 倾听北京声音