TiD 2017
质量竞争力大会

主流研发团队的工程效率与质量安全提升实战

韩葆，
15210834682,Bao.Han@synopsys.com

目录

为什么要做工程效率？- 环境需求

不断演变发展的环境
需要新方法

**新的技术堆栈
和攻击面**

嵌入式设备

云服务（私有云、
混合云、公有云）

语言和框架

**新的开发理念
和方法**

敏捷

开发运维

CI/CD

开源

不断变化的部署环境
改变了测试需求

为什么要做工程效率？- 安全态势

漏洞深深隐藏

- 即使采用先进的工具和方法也很难发现
- 代码或配置很小的改动会产生新的安全漏洞

任何时间 – 永久风险

远程攻击

- 网络访问可从世界任何地点随意发起攻击
- 很难跟踪
- 无法指控

任何人 – 独狼或国家

自动攻击

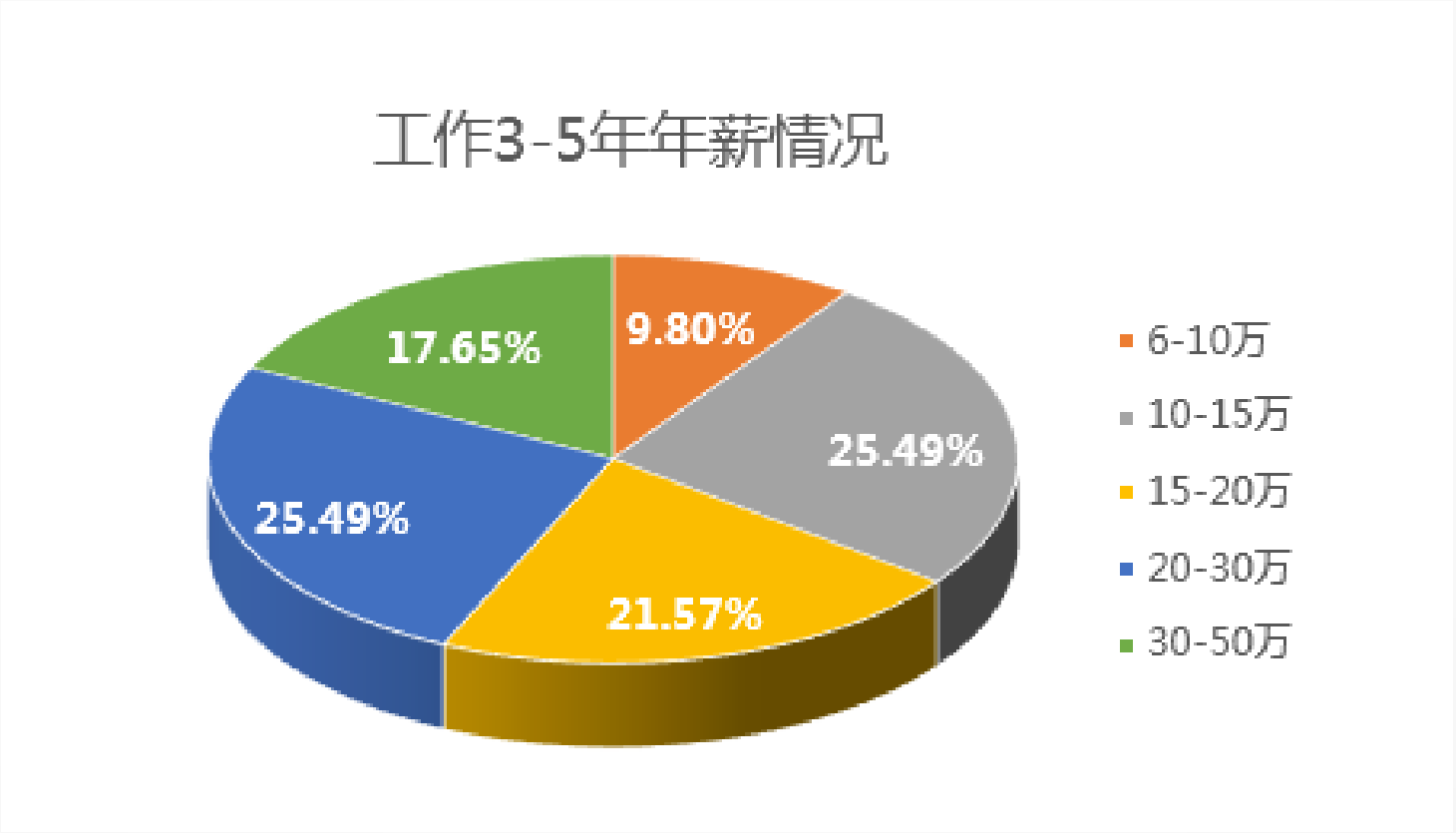
- 广泛共享软件中的一个漏洞可在同一时间随处被用来自动进行攻击
- 示例 – 城市中所有交通信号灯同时失效

大范围 – 大规模攻击

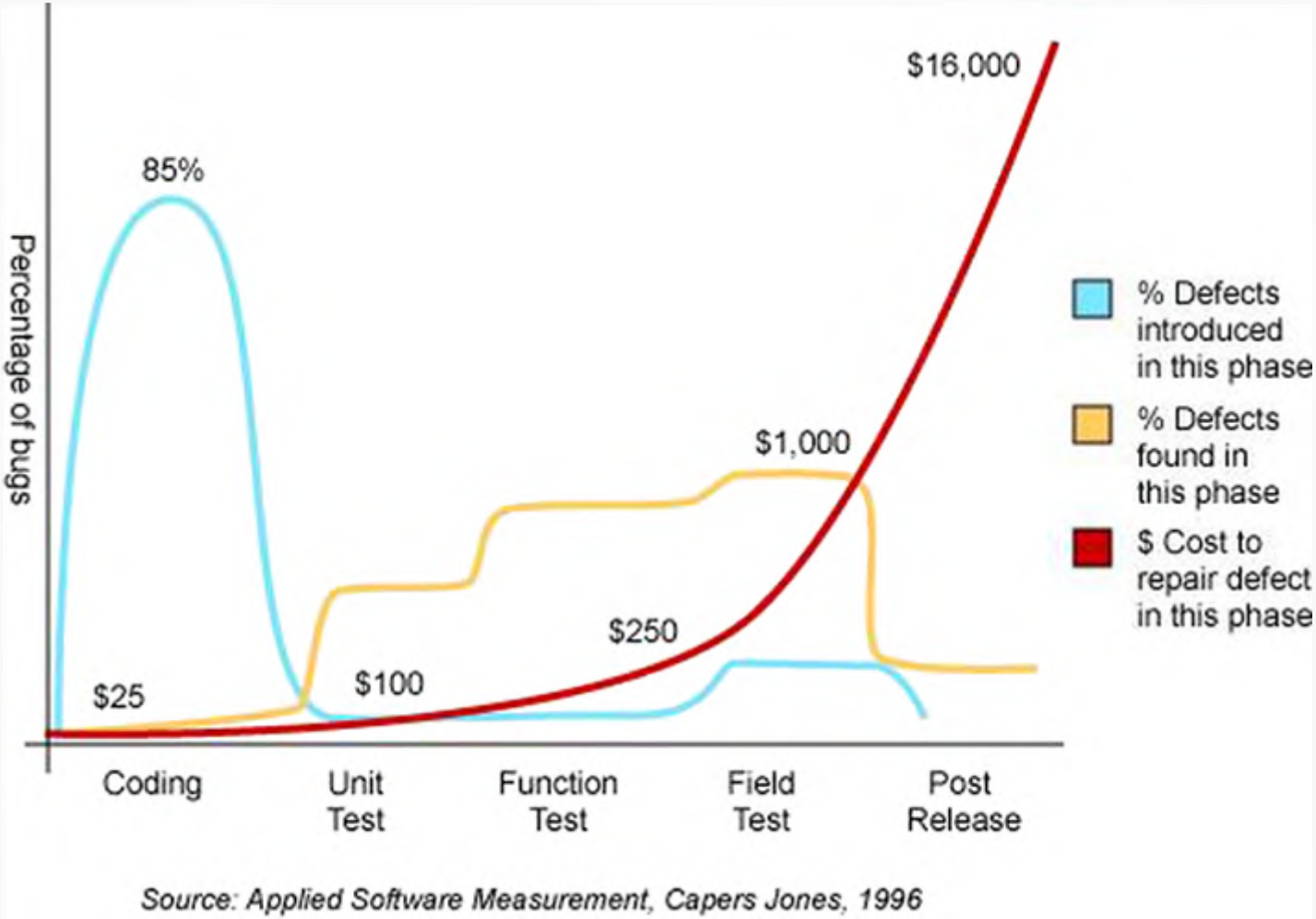
为什么要做工程效率？-人力成本

	Java	PhP	Android	.Net	平均工资
上海	12340	12550	11900	10650	11860
北京	12910	14220	12100	10990	12555
杭州	11920	10910	9800	9120	8188
广州	9700	8750	8980	8200	8907

为什么要做工程效率？-人力成本



为什么要做工程效率？-人力成本



Source: Applied Software Measurement, Capers Jones, 1996

为什么要做工程效率？-管理需求

- 大规模研发团队管理的需求（1000-50000+）
- 集中式报告-质量、安全、性能
- 全系统流程管控
- 资源节省

工程效率平台的典型架构与相关技术

TiD2017 工程效率平台典型架构

项目管控	研发管理	编译与发布
• 需求 • 流程 • 报告	• 源代码 • 集中编译 • 研发测试	• 集成/发布 • 性能测试 • 线上监控

工程效率平台典型架构-纯安全集合

项目管理

- 项目工作流

研发安全

- 代码安全审计
- IAST

安全测试

- 渗透测试

线上监控

- 安全管理与态势感知

工程效率平台典型架构-纯质量集合

项目管理

- 项目工作流

研发质量

- 代码质量- Code Reveiw

功能测试

- 功能、压力测试

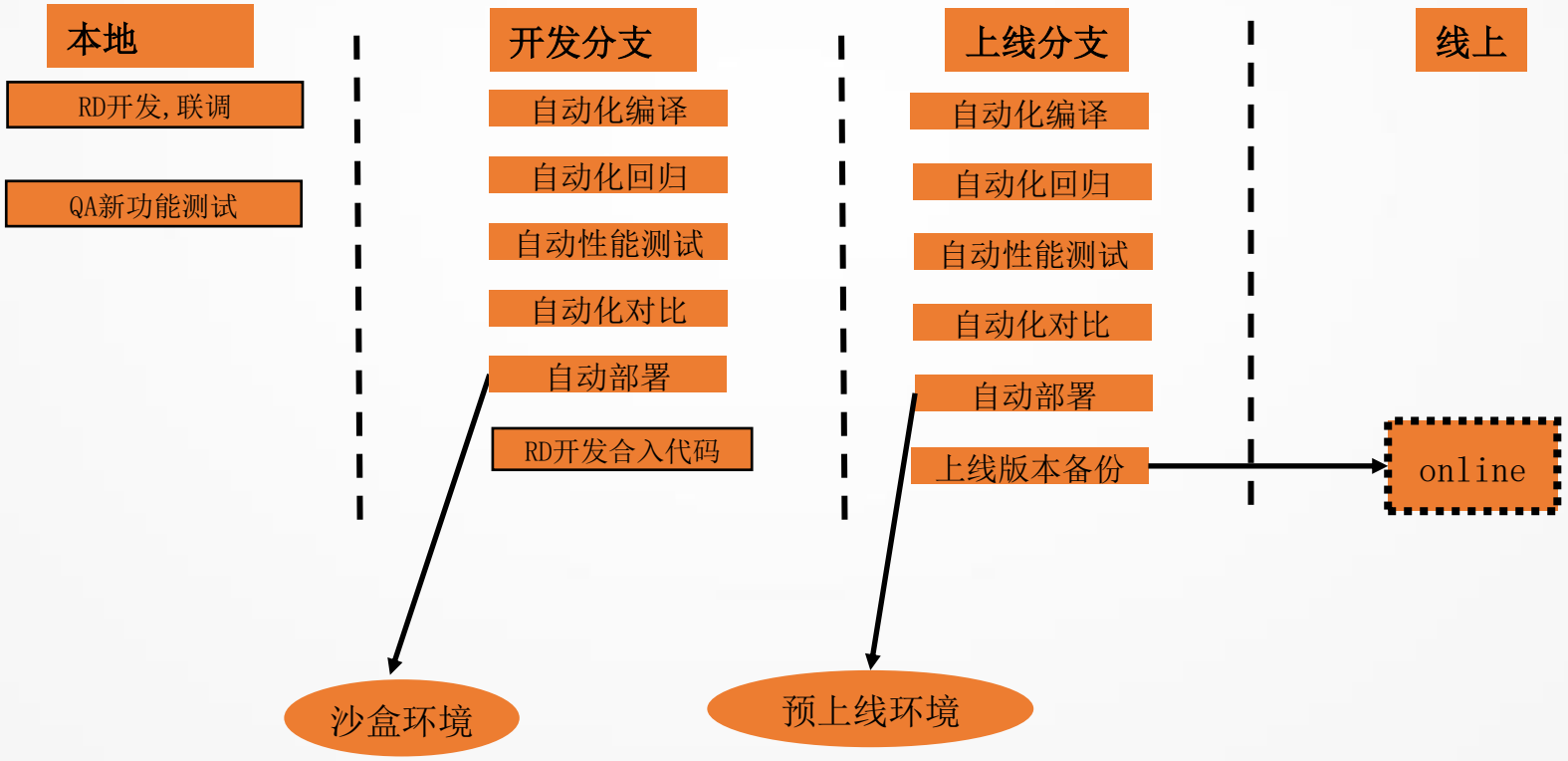
线上监控

- 线上性能、功能监控

TiD2017 工程效率平台典型架构（百度）

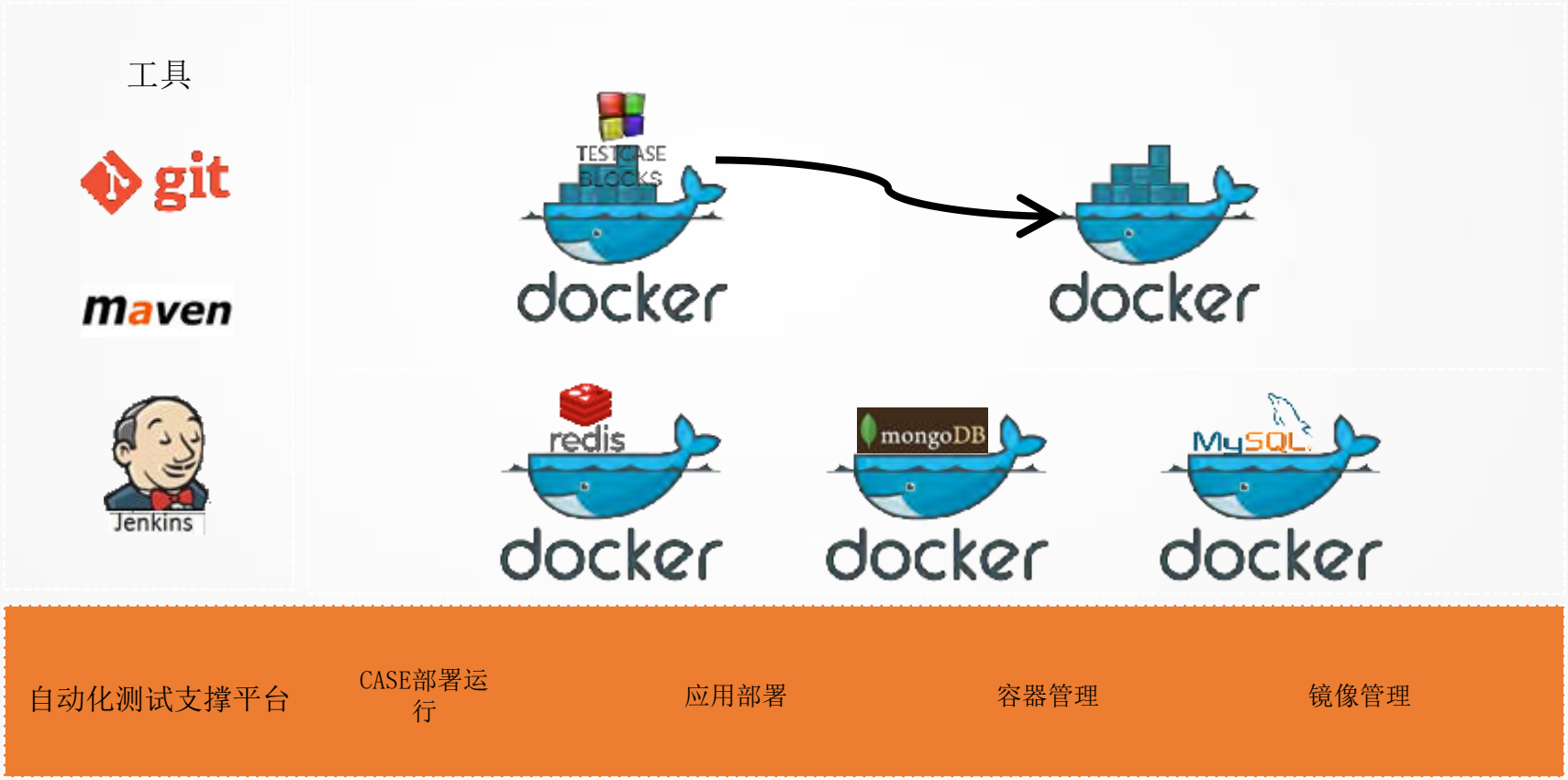
来源：
http://pstatic.geekbang.org/pdf/5718389de9ede.pdf?e=1500283199&token=eHNJKRTldoRsUXOuCP9M3icEhpbyh3VF9Nrk5UPM:UiEvE3mVO3UwuWctXTqqWeAtnbk=

工程效率平台典型架构（链家）



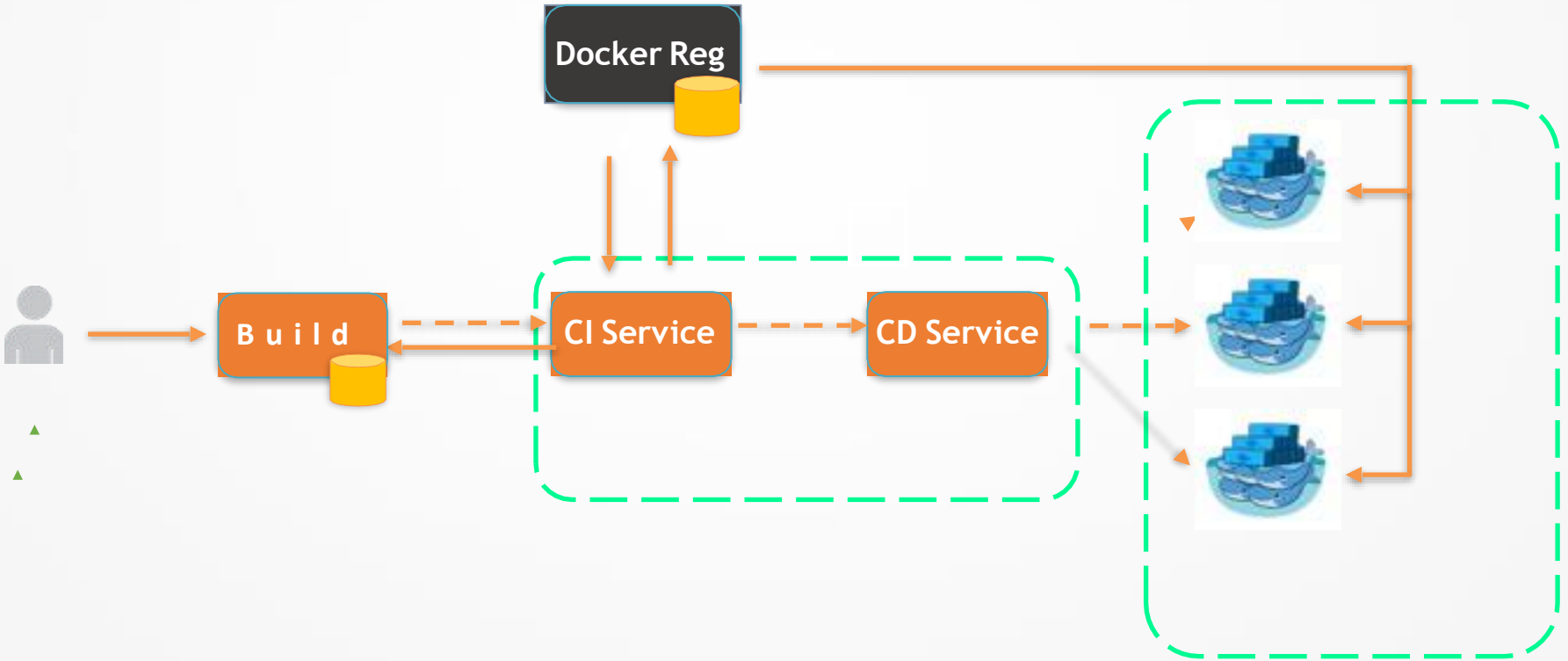
来自：
https://testerhome.com/topics/8910

工程效率平台典型架构（链家）



来自：
https://testerhome.com/topics/8910

工程效率平台典型架构（Cloud CI/CD）



工程效率平台典型技术 - 项目管理

- 需求
- 设计
- 研发管理
- 缺陷管理
- 报告综合
  - 禅道
  - IBM Rational
  - Atlassian
  - …

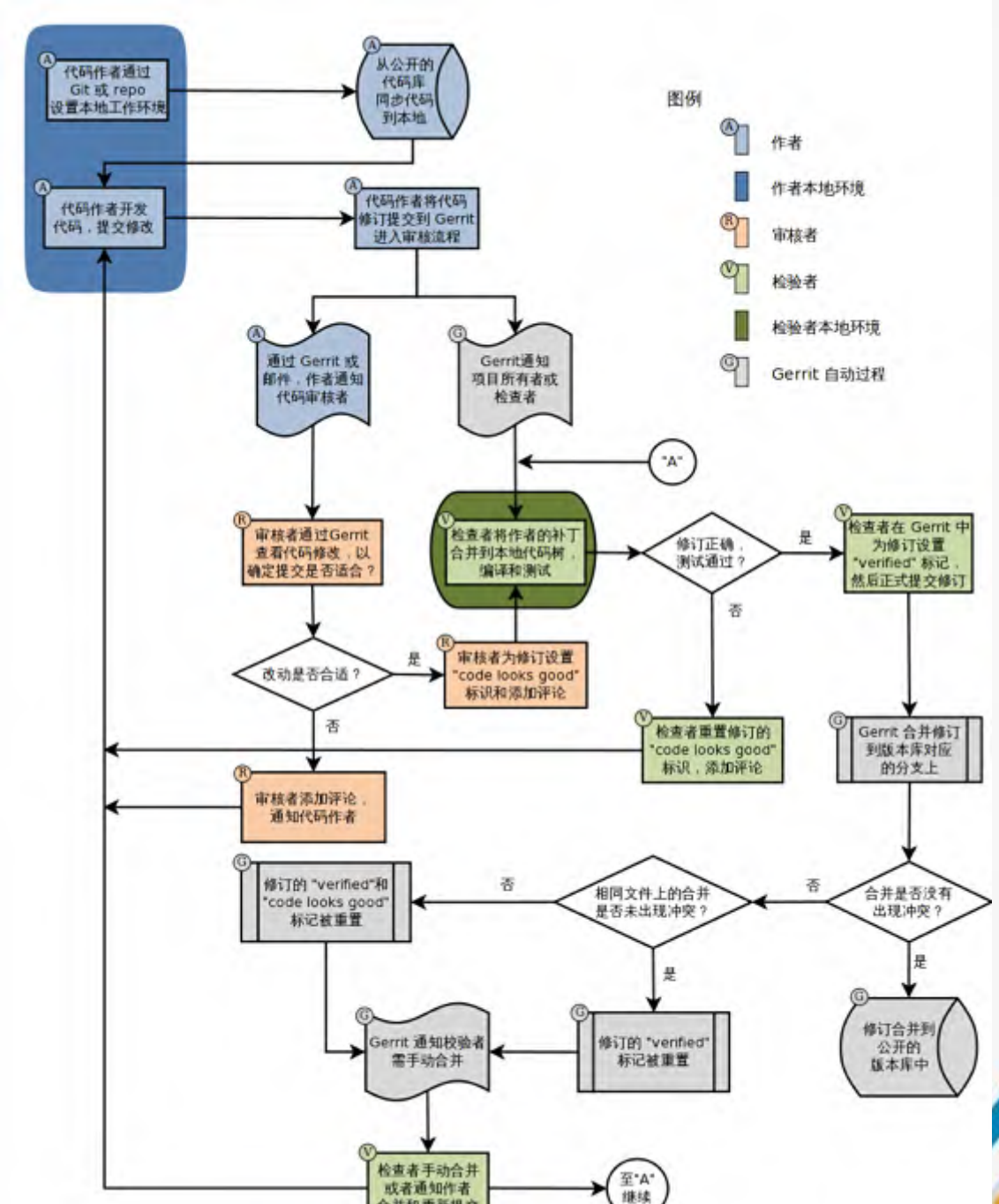
SDLC软件研发生命周期中的技术

工程效率平台典型技术 - Code Review

- 软件开发过程中对源代码的系统性检查
- 查找系统缺陷，保证软件总体质量
- 轻量级代码评审
- 人工/自动化两种方式
  - 人工 - Gerrit
  - 自动化- 代码静态分析

TiD2017

GERRIT



Coverity的静态分析方案

- 基于Meta Compilation的静态分析：
  - 由斯坦福大学教授Dawson Engler提出， 在深度理解代码与程序语义的基础上检测缺陷
  - 旨在查找“真正的代码缺陷”
- 实现原理：
  - 使用可扩展的metal语言定义正确性Checker
  - 将程序的源码使用状态机进行抽象描述（State Machine Abstraction），可执行路径级别分析。
  - 使用xgcc系统匹配Checker与抽象状态机状态，找到问题所在的点。
- 可准确检测实际的Bug（内存和指针问题、资源泄露、缓冲区溢出，数组越界，心脏出血漏洞...）
  - 能够检测高达亿行级别的代码库，避免“状态爆炸”
  - 使用模型检验与符号执行技术，误报率降低至15%以下
- 算法已产品化-Coverity
  - 面向企业的Coverity 软件
  - 面向开源代码的Coverity SCAN

静态代码分析 Coverity®

从源头保证安全

静态分析技术能够在编写代码的同时发现其中的严重安全漏洞。

Synopsys提供Coverity静态代码分析工具

“开发人员首先” 保证安全

- 签交前发现并修复漏洞

企业就绪

- 企业级报表和仪表板
- 超大型代码库 (10M以上LOC)

符合标准

- OWASP Top 10, CWE Top 25
- 报告包括 PCI-DSS

手机安全

- Android和 iOS Objective-C
- OWASP Mobile Top 10, JSSEC, CERT

TiD2017

静态代码分析
Coverity®

从源头保证安全

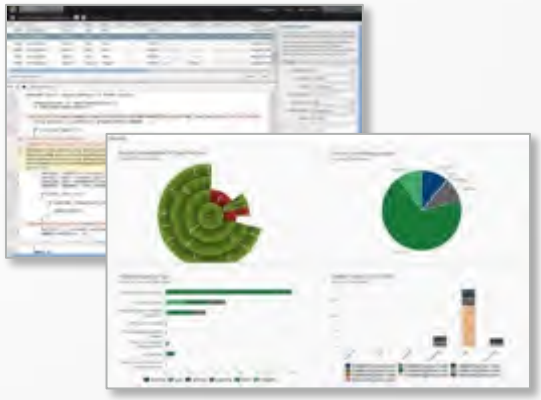
采用行业最高效、最可靠的静态分析解决方案，在编写代码的同时发现其中的严重安全漏洞。

可发现的缺陷包括

- 缓冲区溢出
- 内存损坏
- 资源泄漏
- 资源争用

OWASP10 + CWE25 覆盖范围包括

- SQL注入
- 跨站脚本
- 敏感数据误用
- 命令注入



Android 应用安全策略

质量问题-Java Based

- **深度理解Android API**
- **性能与崩溃类问题：资源泄露，空对象引用，ANR...**

Coverity 代码安全性分析

- **权限控制**
- **泄露分析**
- **....**

TiD2017

Android Security CWE Coverage

CWE	CWE Name	Coverity Static Analysis Checker
259	Use of Hard-coded Password	HARDCODED_CREDENTIALS
296	Improper Following of a Certificate's Chain of Trust	BAD_CERT_VERIFICATION
297	Improper Validation of Certificate with Host Mismatch	BAD_CERT_VERIFICATION
299	Improper Check for Certificate Revocation	BAD_CERT_VERIFICATION
321	Use of Hard-coded Cryptographic Key	HARDCODED_CREDENTIALS
327	Use of a Broken or Risky Cryptographic Algorithm	RISKY_CRYPTO
328	Reversible One-Way Hash	RISKY_CRYPTO
337	Predictable Seed in PRNG	PREDICTABLE_RANDOM_SEED
759	Use of a One-Way Hash without a Salt	WEAK_PASSWORD_HASH
760	Use of a One-Way Hash with a Predictable Salt	WEAK_PASSWORD_HASH
798	Use of Hard-coded Credentials	HARDCODED_CREDENTIALS
916	Use of Password Hash With Insufficient Computational Effort	WEAK_PASSWORD_HASH

服务器端（Java/Javascript/Node.JS）

质量问题/安全问题-Java Based/JS/Node.JS

- NULL_REFERENCE
- COPY_PAST_ERROR
- XSS
- DOM_XSS
- .....

软件成分分析 ProteCode

了解软件构成。

识别第三方软件组件中的已知漏洞并修复，避免被利用。

Synopsys提供ProteCode软件组成成分分析工具

软件组成分析包括：

- 任何软件或器件的完整物料清单
- 企业工作流集成
- 可配置CI插件的开放式API

覆盖范围包括：

- 源代码和二进制代码（包括容器、固件和虚拟HD）
- 开源代码许可义务
- 第三方代码已知漏洞

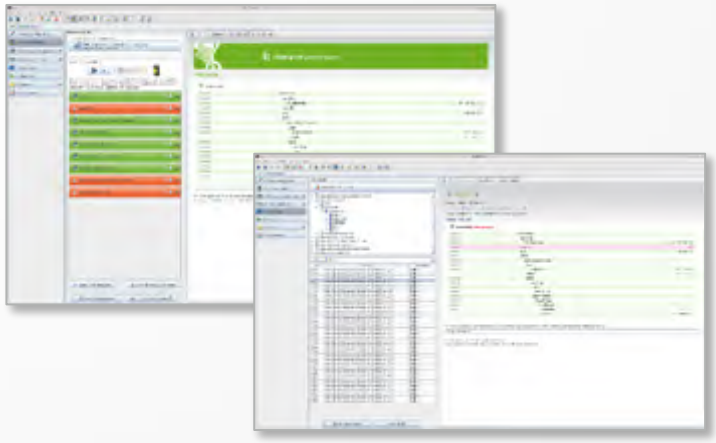
智能模糊测试

Defensics

在黑客动手前… 在软件中发现未知危险漏洞。
Synopsys提供Defensics模糊测试工具

先进的模糊测试

- 降低总体开发成本和时间
- 避免召回/修补/更新造成损失
- Heartbleed漏洞的发现原理-Defensics工具



交互式应用安全测试（Seeker）

将动态测试和运行时代码分析集成到现有开发生命周期之中，帮助开发团队发现并确认多层 web 应用中的安全漏洞

介于白盒测试与黑盒测试之间的安全漏洞检测

- 通过模拟漏洞利用和数据分析，精确评估每一个漏洞风险的影响和分类
- IAST的技术能够覆盖到每一个漏洞的源代码
- BSIMM推荐的的IAST软件工具为Seeker，来自以色列

工程效率平台典型技术 - 其他

- A/B测试
- 压力测试
- 集成测试
- 线上监控
- 态势感知
- …

工程效率平台典型技术 - 工作流集成

工作流集成代表全生命周期集成

落地-工作流集成、培训与实施

工作流集成代表全生命周期集成

培训与指标制定

- 培训设计
  - （线上/线下）
  - 培训指标设定

- 指标设定
  - 千行代码缺陷率
  - 高等级缺陷数
  - 安全漏洞数
  - 缺陷清零

SYNOPSYS®
Silicon to Software™

Thank You

个人微信号