

软件是怎样开发的

后端、H5、微信 Mini 开发实战

by Shane Yoo

2020 年 8 月
in Internet Team

目录

- 一、规划
- 二、需求
- 三、设计
- 四、开发
- 五、测试
- 六、运维

一、项目规划

- 选择适合项目的开发模式
 - 瀑布（周期长、发布烦、变更难）
 - 敏捷（快速迭代、持续集成、拥抱变化）
- 项目计划
 - 工作分解结构（Work Breakdown Structure, WBS）
- 项目管理工具
 - MS Project
 - Jira
- 文档（word、excel、ppt?）还是 markdown
- 风险管理
 - 案例：拼多多“无门槛优惠券”（拼多多错误发布了一个测试用的代金券，用户可以领取 100 元无门槛代金券）
 - 案例：Bilibili 后端代码泄漏（注释）
 - 案例：携程在日志中泄露用户银行卡信息
 - 案例：CSDN 将用户密码明文存储到数据库中

瀑布模型



敏捷模型

敏捷软件开发宣言

我们一直在实践中探寻更好的软件开发方法，
身体力行的同时也帮助他人。由此我们建立了如下价值观：

个体和互动 高于 流程和工具

工作的软件 高于 详尽的文档

客户合作 高于 合同谈判

响应变化 高于 遵循计划

也就是说，尽管右项有其价值，
我们更重视左项的价值。

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

敏捷实践

- Scrum + 极限编程 + 看板
 - 快速迭代
 - 渐进式的架构设计
 - 自动化测试
 - 持续集成
 - 用户故事（子功能、Bug）
 - 小团队
 - 组织扁平化
 - 每日站会
 - Ticket
 - 分工明确
 - 专业精
 - 拥抱变化

问题1

- 你所在项目中，采用的开发模型？

二、需求

- 需求分析：用户需求变成产品需求
- 原型设计：画产品的结构图（功能） & 流程图（联系）
- 工具
 - 需求管理工具（如 Jira）
 - 文档管理工具（如石墨文档）
 - 原型设计工具（如墨刀，XMind，ProcessOn）

问题2

- 该项目的用例图?
- 画出该项目的产品结构图和流程图?

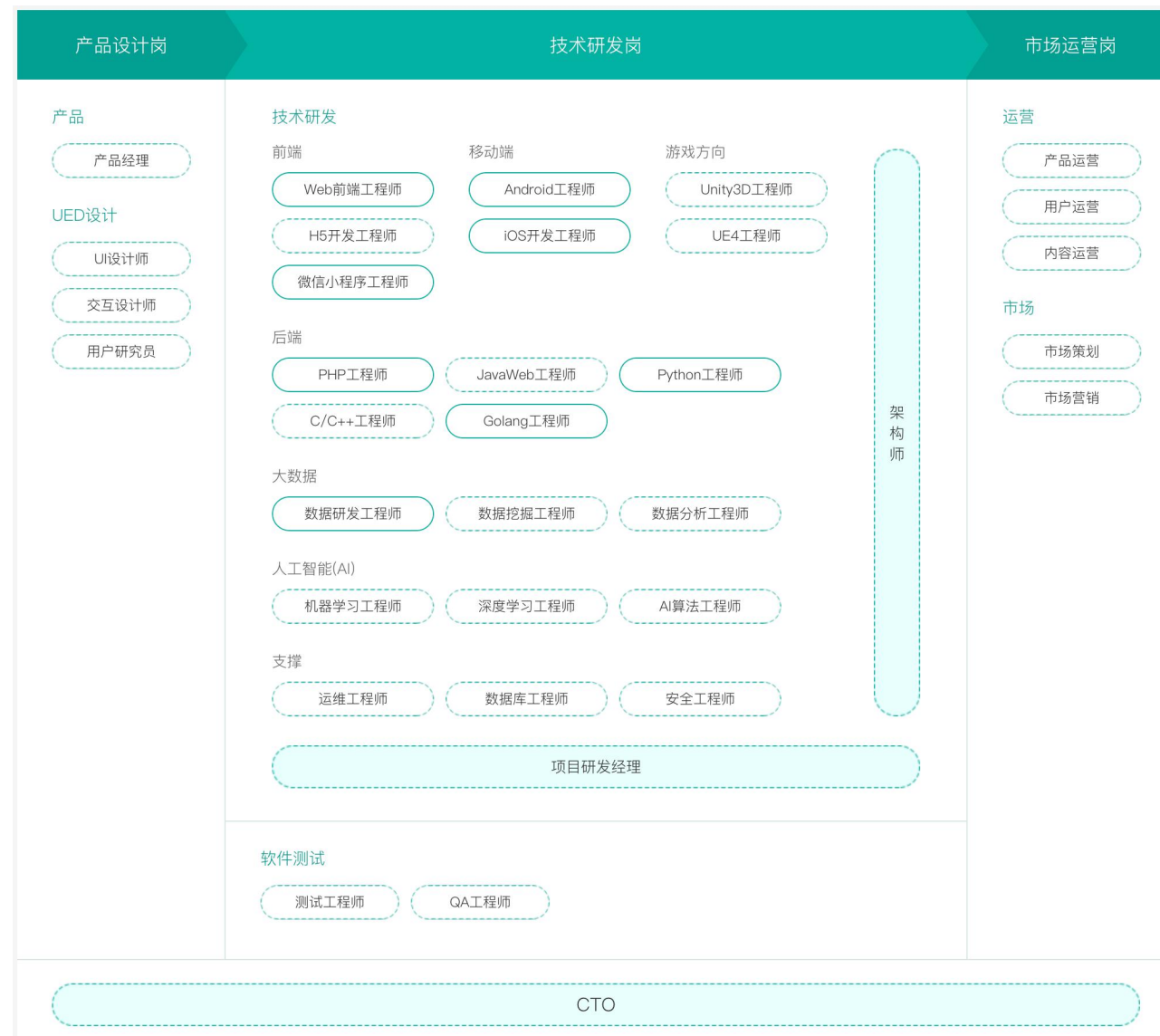
三、设计

- 架构设计
 - 软件架构入门 (<http://www.ruanyifeng.com/blog/2016/09/software-architecture.html>)
 - 图解从单个服务器扩展到百万用户的系统 (<https://mp.weixin.qq.com/s/wSCeO8QVYniMIGcBFDZyjwt>)
- UI 设计
 - 7 steps to become a UI/UX Designer (<https://www.linkedin.com/pulse/7-steps-become-uiux-designer-manu-mehra-ui-ux-designer/>)
 - Iconfont (<https://www.iconfont.cn/>)
 - UI Movement (<https://uimovement.com/>)
 - CodePen (<https://codepen.io/>)
 - UI Tool: 蓝湖、PS、Sketch等
 - UI Library: Bootstrap、Element、ColorUI 等
- API 设计 (trade-off: 问题、技术、风格、定义)
- 数据库设计 (表设计)

问题3

- 分析你所在团队的技术选型?
- 该项目采用什么架构? 尝试画出架构图和部署方案图?
- 该项目的 UI 设计?
- 该项目的接口设计?
- 该项目的数据库设计?

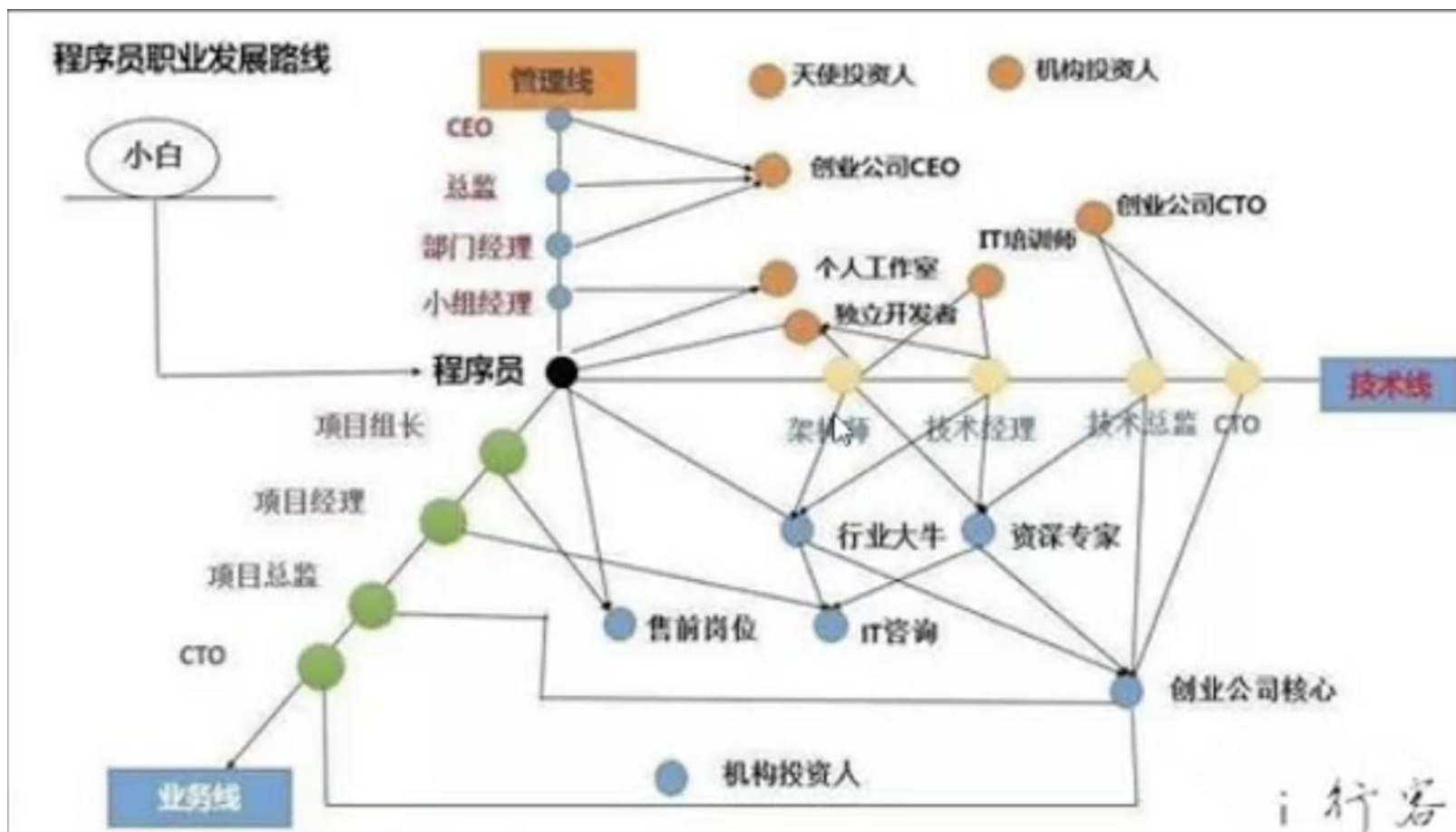
四、开发



程序员职位

- 大前端 (Web, H5, Mini, 桌面应用)
 - HTML/JS/CSS
 - Vue/React/Angular
- 移动端 (Android, iOS)
- 后端 (后台, 接口等)
 - Java
 - Python
 - Node.js
- 算法 (串, 表, 树, 图, 排序, 查找等)
- 云计算 (IaaS, PaaS, SaaS, FaaS, DaaS)
- 大数据 (计算 + 存储 + 应用)
- 人工智能 (数学 + ML + DL)
- 架构 (高性能 + 高可用 + 可扩展)

程序员职业



职业生涯的第一阶段：五年左右时间，把基础打牢

后端开发工程师

- 技能

- Java、Spring Boot
- Vue
- Element UI
- MyBatis
- Redis
- PostgreSQL

- 资料

- Java教程 (<https://www.liaoxuefeng.com/wiki/1252599548343744>)
- Spring Boot Guide (<https://github.com/Snailclimb/springboot-guide>)
- Vue.js 官方教程 (<https://cn.vuejs.org/index.html>)
- Element UI 官方教程 (<https://element.eleme.cn>)
- MyBatis (<https://mybatis.org/mybatis-3/zh/index.html>)
- Redis 教程 (<https://www.runoob.com/redis/redis-tutorial.html>)
- PostgreSQL 教程 (<https://www.runoob.com/postgresql/postgresql-tutorial.html>)

H5 开发工程师

- 技能
 - HTML/JS/CSS
 - Vue/Vue Router/Vuex/Axios
 - Vant
 - Http
- 资料
 - MDN JavaScript 教程 (<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>)
 - Vue.js 官方教程 (<https://cn.vuejs.org/index.html>)
 - Vue Router 官方教程 (<https://router.vuejs.org/zh/>)
 - Vuex 官方教程 (<https://vuex.vuejs.org/zh/>)
 - Axios中文网 (<http://www.axios-js.com/>)
 - Vant 官方教程 (<https://youzan.github.io/vant/#/zh-CN/home>)
 - MDN HTTP (<https://developer.mozilla.org/zh-CN/docs/Web/HTTP>)

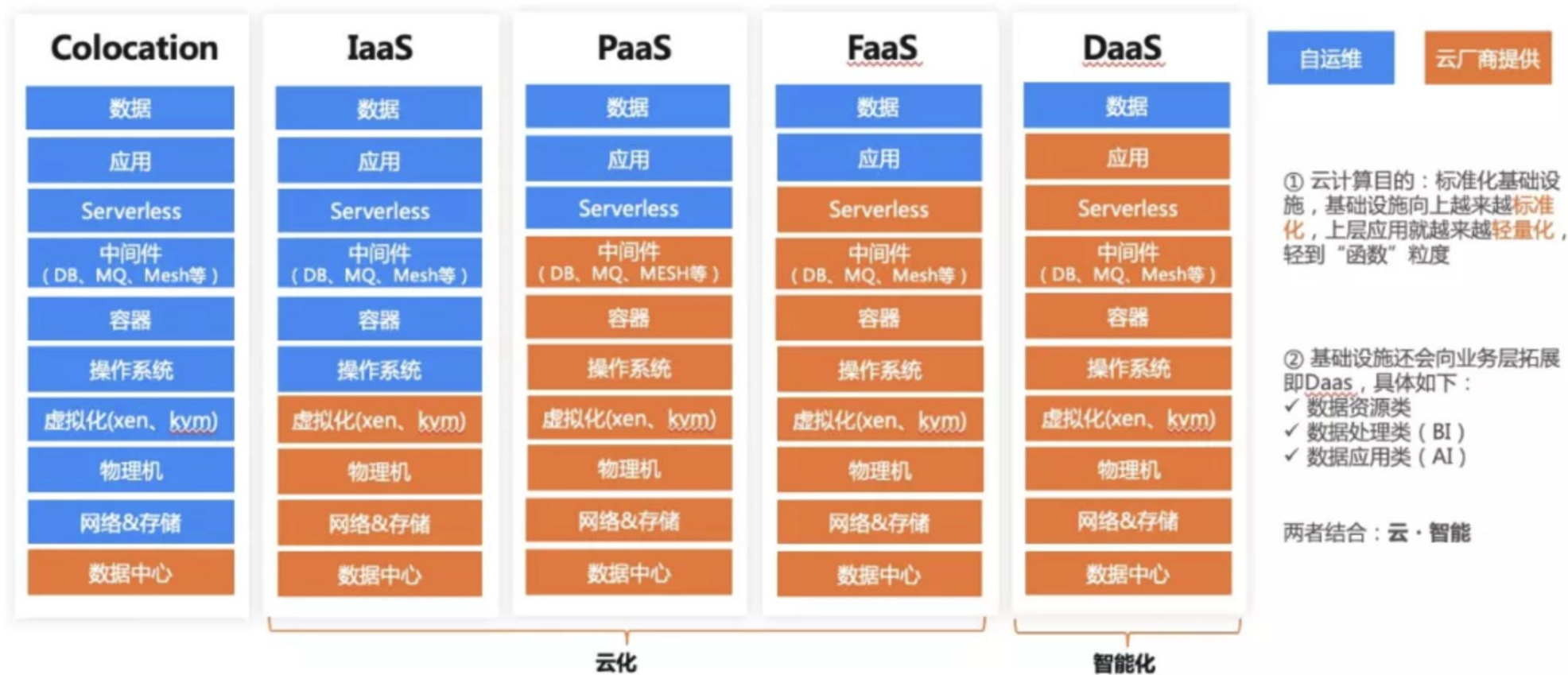
Mini 开发工程师

- 技能
 - HTML/JS/CSS
 - 微信小程序开发文档
 - Color UI
 - Http
- 资料
 - JavaScript 教程 (<https://www.liaoxuefeng.com/wiki/1022910821149312>)
 - 微信小程序开发文档
(<https://developers.weixin.qq.com/miniprogram/dev/framework/>)
 - Color UI (<https://github.com/weilanwl/ColorUI>)
 - 图解 HTTP (<https://book.douban.com/subject/25863515/>)

全栈工程师

- 核心：Web 技术
 - 网络
 - 前端
 - 后端 MVC
 - 持久化技术
- 后端进阶
 - Backend Developer (<https://roadmap.sh/backend>)
- 前端进阶
 - Frontend Developer (<https://roadmap.sh/frontend>)

云管端之云1



云管端之云2

- 资料
 - 云原生已来，只是分布不均
(<https://mp.weixin.qq.com/s/apihIHmPwyVdZZi5d7U0Wg>)

云管端之管1

- HTTP (方法、状态码)
- HTTPS (HTTP + SSL/TLS, 加密技术)
 - TLS 是如何加密的?
 - 证书是如何加密的?
- 服务端推送技术 (Pull -> Poll -> 假Push -> 真Push)
 - Comet
 - WebSocket
 - HTTP/2
 - WebRTC
- SOAP & REST
 - 特点
 - 技术: soap+xml、HTTP + JSON (REST)
 - 适用场合

云管端之管2

- 工具 (telnet、netcat、tcpdump、Wireshark、curl、jq)
- 资料
 - MDN HTTP (<https://developer.mozilla.org/zh-CN/docs/Web/HTTP>)
 - HOW HTTPS WORKS (<https://howhttps.works/>)
 - HTTP Cats (<https://http.cat/>)
 - HTTP STATUS DOGS (<https://httpstatusdogs.com/>)
 - HTTP访问控制 (CORS)
(https://developer.mozilla.org/zhCN/docs/Web/HTTP/Access_control_CORS)
 - SOAP 教程 (<https://www.w3cschool.cn/soap/>?)
 - REST 和 SOAP: 谁更好, 或者都好?
(<https://www.infoq.cn/article/rest-soap-when-to-use-each>)

云管端之端

- 资料:
 - 公众号、H5、小程序、APP 比较
(<https://mp.weixin.qq.com/s/sAGl4eRo82ejmICH0MxmMQ>)

开发工具

- 笔记本（如Mac, Linux, Windows）
- 开发工具（如IDEA, WebStorm, DataGrip, VS Code, Vim）
 - 资料：再见，Navicat！同事安利的这个IDEA的兄弟，真香！
(https://mp.weixin.qq.com/s/FGR7ObhuEpiWN8_ogxDdSQ)
- 源代码管理工具（如git, svn）
 - 资料：Git 工作流程
(<http://www.ruanyifeng.com/blog/2015/12/git-workflow.html>)
 - GitHub：全球最大同性交友网站
(<https://zhuanlan.zhihu.com/p/33991691>)
- 持续集成平台（如Jenkins）

开发部署

- 手动部署
- 自动化部署 (脚本部署)
- 持续交付

程序员的自我修养

- 反例：一年经验用了三年甚至更多
- 好的软件开发工程师的成长过程：
 - 软技能的成长
 - 业务能力的成长
 - 专业技能的成长

软技能

- 学习能力
- 时间管理能力
- 逻辑
- 系统
- 踏实
- 合作
- 分享
- 审美
- 心力

业务能力

- 产品思维：挖掘痛点和真实需求
- 运营思维：数据分析与用户行为的洞察

专业技能

- 快速学习能力
- 解决问题能力 (搜索)
- 专注某个领域
- 算法
- 编程
- 知行合一
- 影响力

架构师

- 反例：PPT 架构师
- 好的架构师：努力（程序员、模仿、学习、行业）
 - 在别人的指导下工作，新手
 - 独立完成模块功能开发，对自己的工作负责
 - 帮助别人解决技术问题
 - 指导他们工作，对团队的工作负责
 - 开发可复用的框架、工具、组件，对整个产品的技术负责
 - 进行大规模系统的架构设计，对整个系统负责
 - 抽象、提炼自己的工作形成文档，总结方法，技术创新，培新新人，为公司和未来负责
 - 领导开发全球级的开源产品（github），通过文章、书籍、演讲，推动技术领域的进步与发展，为行业与整个技术界承担责任

问题4

- 配置该项目的开发环境
 - Java
 - PostgreSQL
 - Node
 - Git
 - IDEA
 - WebStorm
 - 微信开发者工具

五、测试

- 软件测试的主要工作：发现 Bug，报告 Bug，跟踪 Bug
- Bug 跟踪工具：Jira
- 安全性测试工具：APPScan 是 IBM 旗下的一款漏洞扫描工具，支持网站和移动 App

问题5

- 为什么有些大厂不需要测试工程师?
- 分析你的团队是否需要测试工程师?
- 什么样的公司需要专职测试?
- 你所在项目中，在安全方面有哪些可以改进的地方?

六、运维

- 软件版本
 - 主版本号 . 子版本号 . 修正版本号
- 软件发布
 - 发布流程: 代码冻结、Bug 分级、回归测试
- DevOps: 持续交付+日志监控+虚拟化技术
- 持续交付
 - 资料: 持续集成是什么?
(<http://www.ruanyifeng.com/blog/2015/09/continuous-integration.html>)
- 日志监控: ELK (Elasticsearch + Logstash + Kibana)
 - 资料: ELK教程 (<http://docs.flycloud.me/docs/ELKStack/>)
- 虚拟化: Docker + k8s

问题6

- DevOps 工程师是做什么的？
- 你所在项目中，对线上故障如何处理？

3W1H

- What
 - 一项目三应用
- Why
 - 沉淀
 - 开发能力
 - 兴趣
 - 团队
- How
 - 初始化三个应用
 - 学习
 - 沟通

一些唠叨

- 拥抱变化，持续学习。
- 因为信任，所以简单。
- 认真生活，快乐工作。

Q & A

Thanks !