2015Qcon 北京参会日记

数科开发部 褚秋实

目录

楔 子

但凡做事特别是和技艺相关的，都会涉及三个境界：见自己，见天地，见众生。

作为入行有些年头的一线程序员参加 2015Qcon 北京全球软件开发者大会可以说是这三方面都得到了 focus 和 update。

在人流嘈动中，你会发现大家都是有着相同的专业背景，他们每天也是干着跟代码打交道的活，你不由得会重新审视自己，回顾自己，思考自己。

看着台上侃侃而谈的人，大家总是泛起对他们的敬佩的眼神。是的，程序员就是这样一群单纯的人，他们尊重知识。在我眼中他们是高山，是我努力的方向，无限风光在险峰。

门里进行着一场场精彩的分享，门外摆满了各家产品、服务的展台。各种技术的，商业的元素交织在一起，整个行业众生的一个鲜活 view 展示在你面前。

软件正在改变着世界，在这个越来越智能的时代，互联网垂直深入各个领域，工程师将有更广阔的舞台，我们唯有不断学习，分享，交流才能跟上潮流，才能 handle 各种新的问题，才能将各种创新的 idea 实现

第一天 敏捷，团队建设，安全，全栈开发

Topic1 异步处理在分布式系统中的优化作用

回顾

讲师是来自阿里巴巴的研究员，专攻网站性能优化，还有海外工作背景，经历了 facebook 各个阶段的优化升级。讲师的一大观点就是强调搞 it 不光是注重工程学方面，而且还要本着科学态度深入钻研，如一个 web 应用如何在提升性能可以深挖很多内容。

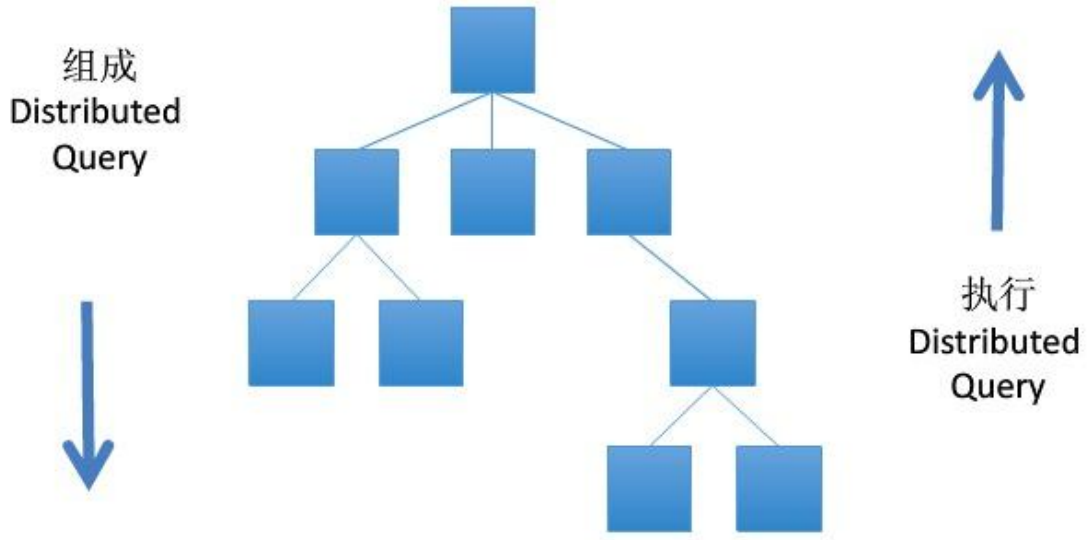
构建异步处理的分布式架构是当前提升 web 性能的一个主要途径，但是异步处理无形中增加了开发的难度，讲师提出了可能需要增加一个额外的层来负责异步查询的优化，让系统支持更加灵活的优化策略。另外讲师还指出由非异步要改成异步处理是一个痛苦的过程，因此在一开始最好就要考虑异步处理的需求，至于如何加这个异步查询优化的层，讲师则没有给答案，只是把这个问题抛给大家。

反思

IT 到底是一门工程学还是一门科学，应该是两者属性都有，在不

同应用场景可能两种比重不一样，所以程序员有时还是要有科学的态度能够进行一些深度的钻研，就这个 topic 而言我觉得要实现在代码层面来看写得很直观，然后在机器层面执行效率又好，这真的是一门艺术啊，可能要从编程语言和开发框架方面多做努力才能更好享受异步处理的好处，作为 FP 的拥趸，我想说函数式编程这方面支持得挺不错

（异步查询树图示）各种异步查询最终在执行的时候构造成一个树，优化契机在此



Topic2 在敏捷中实现持续改进之必要性

回顾

讲师是外国人，一个独立的敏捷顾问，由于语言原因讲师的很多内容不是听得很清晰，但是讲师倡导的主要的理念其实在我们公司自己

敏捷实践中早已经用到了，参会的很多其他公司的人都还表示之前没有了解到这些东西，由此可见我们的敏捷转型还是走在前列的。

讲师主要的一线观点随堂记录如下：

Real agile need continues improvement

Find your own best way

Early feedback

Skilled people with method and tools

Right person do right task

Depends on team

反思

持续改善，尽早反馈….种种敏捷原则我已经难熟于心，但是现实在团队中如何实践好才是困难的地方，所谓知行合一，我想只有“用心”了，从细节每一步不断尝试，不断获取反馈，不断改善。 ps 会上有几个人能够和外国讲师很流程的英文交流，真心佩服，所以语言技能很重要

Topic3 Scala vs Clojure

回顾

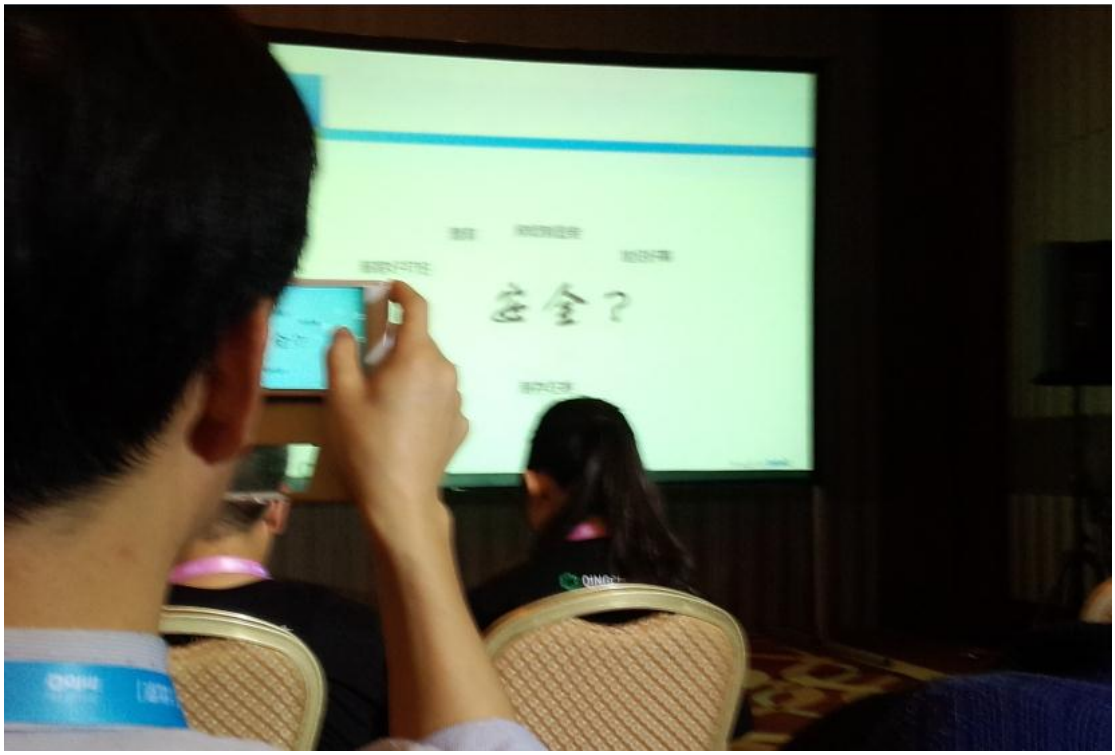
这场也是一个老外讲的，我及时拿了一个同声传译设备，但是效果也不好，内容主要是对比这两种 JVM 的动态语言，讲师是很 open

的，全是对比两种语言吐槽，基本都是在对比两种语言在什么地方做得好，还是不好，各种吐槽

反思

能够吐槽一个编程语言不好，指出其缺点，必定是对此语言有深入的了解，讲师思路还是很对的，主要从代码库，社区，文档，工具等方面来对比吐槽，其实这个思路就是我们做开源选项的思路，不求百分之百满足我们需求，但求解决我们百分之八十的核心需求。

Topic4 让安全飞-新时代下的安全产品思考



回顾

讲师是来自腾讯安全总监，一来就讲了开房数据泄露（很奔放），

阐述自己的了安全观，然后是腾讯安全相关炫技。腾讯的业务种类很多（是啊什么都做）所以内部没有统一的开发框架，很自由的让业务，让产品自己选择开发技术，因此加大了安全管控的难度，但是，但是他们还是在困难中建立了腾讯安全的“宙斯盾”，该系统集扫描，检测，分析，防护，估损一体，很强大的，**不明觉厉**（不是很明白是什么，但是觉得很厉害）

反思

安全问题很容易就成为致命的问题，所以现在安全俨然成为刚需了，自然安全专场的讲座是爆满的，我们到**底是搞安全呢，还是被安全搞**，讲师说得好要建立安全防护统一民族战线，团结开放，运维，业务等一起对抗不安全因素。

Topic5 全栈工程师的自我修养

回顾

讲师是来自百姓网的工程师，首先就明确了全栈应该如何正确理解，全栈不是指会各种技术栈，全栈的全是全在在单个工程师能够有能力覆盖产品生命周期第一期（特别强调产品初期），完成idea->requirements->dev->opt,所以讲师所说的全栈至少是 PM+程序员。靠谱的全栈工程师对企业的价值在于能够独立快速完成初期产品idea的实现和上线,可以这样可以降低成本,如果idea上线不错，则全栈模式退出，各种角色组成团队跟进，如果不幸idea 上线不成功，即使舍弃带来的损失也不大。另外，全栈还有一个好处就是无形中降低了沟通成本。

反思

人力成本是软件产品的主要投入，所以就想到了让一个人快速搞定整个流程的事情，在互联网环境下这可能还真是趋势。筒子们，加油学习吧打造自己的技能金字塔，注意要有最精通最擅长的领域，然后还要附带掌握另外其他技能。现实来说打通了前端，后端，移动端三端的技术还只有 JavaScript，在技术实施上 js 是个可行的全栈开发选择

Topic6 打造高质效的技术团队

回顾

讲师是来自阿里的团队主管，演讲内容都是管理方面的干货，讲师总结了国内开发团队的一些常见问题，如职业水准偏低，自我管理差，基层技术管理缺失等，这些问题严重影响团队工作质效的可持续

提高，然后讲师分享了他的实践，基层技术管理者需要选准切入点，最好从代码入手，通过实干和帮助获得团队的认可和信任，使自己有话语权和影响力，然后制定规范保证团队的纪律性，以集体的力量约束个体，让大家养成好的工作习惯。还要促进知识管理，打造开发的工作环境，增强大家的能动性。

反思

‘基层技术管理者’，这个概念提得好，应该是有必要强化，清晰这个角色。其实在每个团队中我相信都可能存在这样的角色，他们在团队中有影响力，有话语权，得到团队成员的信任，所以这个角色应该要学习如何让团队更高效可持续运作

第二天可扩展、高可用架构设计，编程语言实践

Topic1 微博平台在大规模、高负载系统中的典型问题

回顾

讲师是新浪微博平台的架构师，很明确主要是讲在大规模，高负载系统中的典型问题，主要分了两类问题性能问题和功能问题，双方会互相影响。引发问题总结主要有业务流量暴增系统过载，还有应用系统新发布引入新的问题，然后着重炫技一下他们的监控平台，各种图形化，链路追踪，节点异常（链路追踪可能更容易发现在那个环节上发现出了问题）。然后讲了问题处理的思路：

观察现象，主要是从日志里面来观察问题，日志要包括业务日志，性能日志，以及容器日志（容器可能是 os 或者应用服务运行的 server），看日志要注意从时间点，请求类型以及级别等方面去快速过滤筛选

查勘现场，主要是指运行时细节分析，如线程快照，调用栈等代码执行层面的分析最好聚合多个维度。

分析原因，主要从应用本身，运行环境，硬件等方向，应用主要可能的原因有死锁，内存溢出，依赖资源挂起等；运行环境主要有虚拟机，kernel，通信协议等；硬件主要考虑设备，网络等，分析原因的时候需要多角色联动 dev opt dba sa 之类的协同。

解决问题，首先要能够复现问题确保分析的原因是正确的，然后在修改上线，当然要保证没有引入新的问题。

然后提了一下灰度发布，具体怎么做灰度发布没有细讲，应急预案也没有讲。

思考

会后我专门跟讲师单独交流过，就应急“救火”处理，以及灰度发布进行了探讨，他们的应急处理有 4 板斧：

遇事先回滚，任何变更（应用层面，基础架构层面等）都需要有完备，可靠的回滚方案

降级，故障发生是优先保障核心功能，快速隔离问题节点，降低故障影响，不管是物理上的还是逻辑上的要事先埋点确保能够在故障发生是能够够得到，隔得了

扩容，确保有 30%-50%的容量 buffer，能够快速实现扩容，所以 docker 火得很有道理嘛（据说 2 分钟扩容 100 台 docker）

流量迁移，说多机房切换也好，灾备环境切换也好，都是一个意

思，防止一个粒度比较高的单点故障

总之这 4 板斧应该足以应对紧急情况，，其实大家的方法基本是相通的。

至于灰度发布的话，他指出必须实际行动去想办法构建灰度环境，同时业务也要有一定容忍度，总之是在**物理层面或者逻辑层面控制上线功能的流量**，他们的开发框架就很好支持特性开关的植入，可以使运维能够很好的在线上做各种切换

Topic2 九个月实现破亿用户的可扩展架构

回顾

讲师是来自美图的架构师，内容可以看到一个典型的互联网应用的架构演进，产品刚开始架构非常简单就是 webserver+mysql，讲师提到刚开始一个能跑起来的系统比一来就设计一个优秀的架构更重要，所以他们**一直在保持架构整体的简单性以适应业务的快速变化**

当然到了用户量上去以后，流量突增，他们在高可用上面也做了架构调整，首先要保证容易扩容，另外尽量使用云服务，并且提到一点就是使用其他云服务的时候也要考虑可用性的保障，他们的实践是**同时使用多家云服务以备份**，另外还有个重点就是核心功能隔离，他们踩的一个大坑就是把重要，不重要的功能都绑在一起，导致故障的时候没法通过降级的策略来保证可用性。

另外还提到一点，对于引入开源的实践，引入开源服务确实是可以快速实现功能，但是要保证高可用的前提下可能需要很强的运维保障，或者深入的全面掌握这个开源服务，如果你没法做到这两点的话，架构上要注意松耦合，让自己服务与引入的开源模块弱依赖，即使引入的开源服务出现了不可靠的情况也不会对自身的可用性造成很大影响。

反思

足够简单变起来才能足够快，所以保持架构的简洁也是比较重要的，但是随着业务的复杂度和系统高可用的需求不断增加，架构必不可少的要引入复杂度，如何管理好这些复杂度更好的支持两者就是问题关键。Keep It Simple, Stupid 也许是不错的选择

Topic3 搜狗广告流式计算实践

感觉讲了一个很复杂的架构，但是这个系统解决什么问题没说明白，一堆各种开源框架的拼盘，感觉挺宏伟的，不明觉厉

Topic4 java 字节码技术及其应用

无干货，讲师讲得倒挺幽默的，广告也很直接

Topic5 指数级增长业务下的服务架构改造

回顾

讲师是来自一个提供 IM 通信云服务的初创公司，也是高可用方面的架构设计，开篇主要是广告来着，然后讲了一些架构通用原则如从水平方向应该考虑容量扩展，垂直方向应该考虑业务独立，另外还有读写分离。需求业务增长以后要考虑服务间解耦，可以通过异步处理来实现服务间通信，或者使用 radius 或者 kafka 等组件。另外也是吐槽了开源的坑，初创公司大量使用开源软件，他们在开源上面也踩了一些坑，他们是对外提供云服务，但是也大量使用别人的云服务，强调是要做好对别人云服务的监控，小心被别人坑了。

反思

所以对待开源的态度架构上应该还是持怀疑态度，在决定使用以后要让自己的服务尽量降低对开源模块的依赖，还有使用第三方云服务也是一样。

现在不是云服务都很 low，云的门槛真的这么低啊，自己开发实现自己业务领域的核心问题其他的交给其他云。天空中的云越来越多越来越厚，它们之间又交织在一起，如果来一个雪崩效应……

Topic6 无用之用—互联网创业公司小众语言的使用

回顾

讲师是来自小米网络的，讲的是小米如何用 GO 语言来构建它们的战略核心系统——秒杀抢购系统，据他介绍秒杀抢购业务在小米被提到战略的高度。但是最初的秒杀系统真的是够单纯，就是 php+mysql（难怪抢不到，老卡死），秒杀业务又特殊，需要精准的时间控制，面对突发海量请求，而且对事物又严格要求，所以最初的设计只是勉强应付，然后就扯出新一代秒杀系统，然后讲一堆为什么用 GO 语言来实现，接着又扯出一系列的小众语言，讲师形容他们是无用之用，实际中可能毫无用处，但是特定环境可以发挥作用，而且各自都暗含先进的编程思想。讲师一直很小心避免挑起编程语言谁好谁坏的争论，最后总结到要结合自身领域特点，选择时候自己的开发语

言不管是工程语言还是小众语言，程序员用着高兴才是最好的生产力。

反思

是的，要避免把自己变成只有一把锤子的人，技术快速变化，不能墨守要跟上才能享受技术的红利

Topic7 基于 Scala 构建响应式流计算

挺失败的演讲，不知道是在讲语言还是在讲架构，听下来不知在解决什么问题，不明也不觉厉，也是是我饿了注意力不集中

第三天 微服务，Devops，敏捷

Topic1 微服务与文化

回顾

讲师是美国人来着，prezi 的创始人，prezi 是一个在线演示 ppt 的互联网应用，打广告以后，他首先说明不是技术发言，主要分享的是微服务的文化，主要从微服务下的人员和组织结构两方面，主要吐槽了巨型大服务系统下分工细致，各种角色界限明显，部门之间各自为政，无法提升开发上线速度。然后讲了微服务文化体制下小团队可以实现自我组织，打破孤岛，并且制造自动工具实现持续集成，持续交付提升交付速度。最后讲师还特别提醒改变团队工作方式的代价可能要大于改变技术，**微服务首先是个文化相关的问题，其次才是技术层面的问题。**

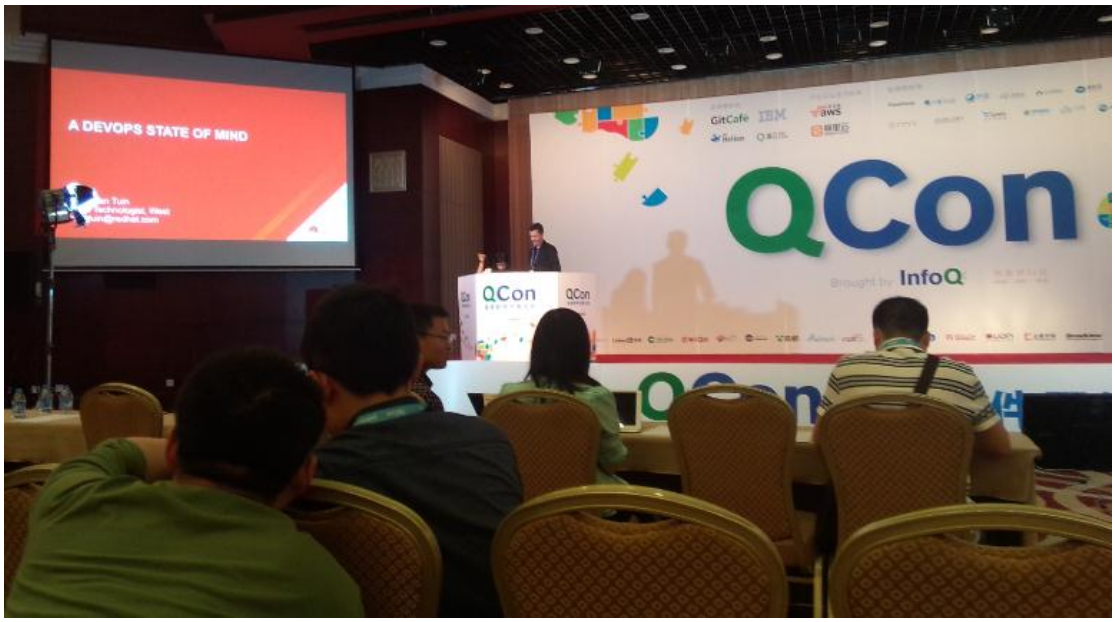
反思

选什么技术，用什么框架真的不是大问题，改变人和组织的工作方式才是挑战，程序员还是要多考虑下人的因素，不要只关注和机器打交道

Topic2 让商业智能成为主导

不知来历的黑人美女打广告，介绍他们的商业 BI 工具如何帮助客户实现价值，一些报表之类的东西，据说低成本 BI 两周见效

Topic3 论 devops 的思维方式



回顾

讲师 redhat 首席技术主管，感觉来头不小，主要讲 devops 的，首先讲到企业级 IT 发展到今天，面临如何让 IT 高效并且降低成本的

问题，所以就有了 cloud，agile，devops 等思路，devops 主要打通开发和运维之间的墙，打通人，流程，技术等以实现 CI 和 CD，只有 IT 高效了才能更好支持企业业务，IT 不能成为瓶颈。

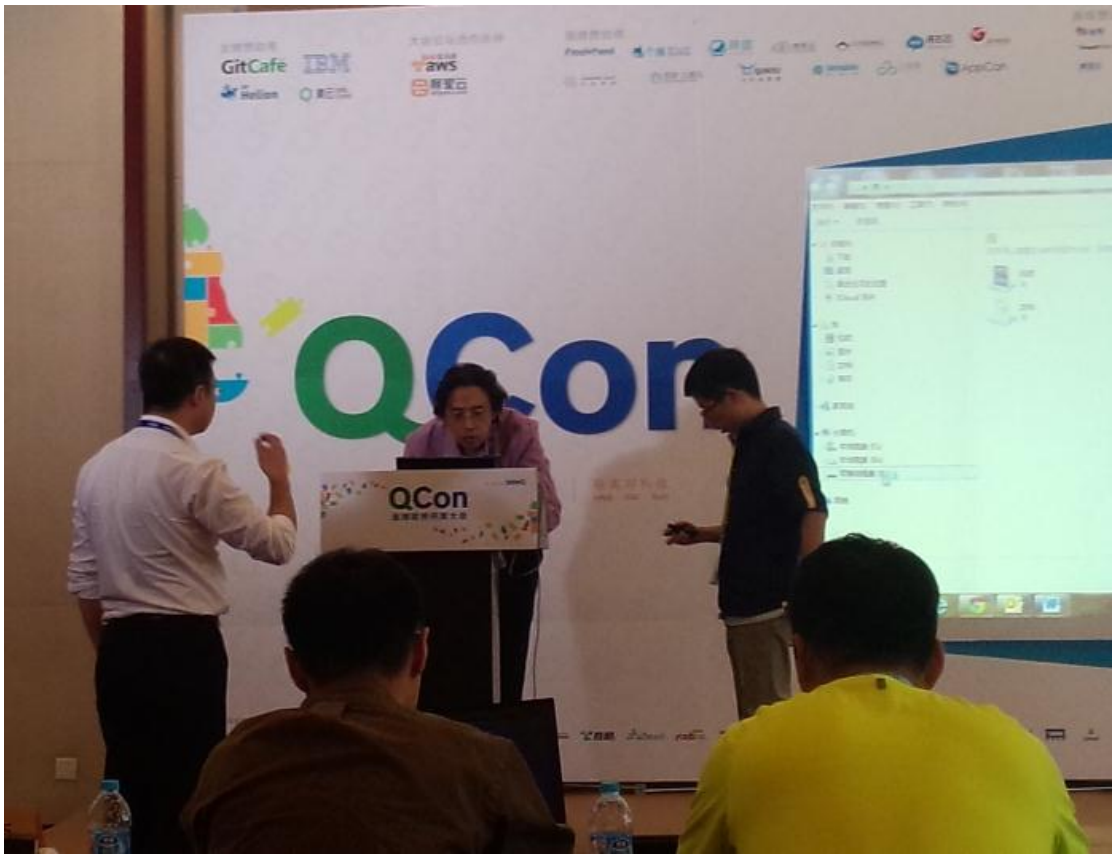
反思

没有什么广告成分，当然 redhat 主打开源生态链的技术炒作就是他们最好的广告，让 IT 高效降低成本真是一个世界范围内的问题，所以 IT 先要变化得快才能支持企业业务变化快。

Topic4 满载价值的敏捷回顾

又是那个荷兰大叔的分享，他的英语口音真的是听不了，主要讲开好回顾会议，打广告他有一本专门讲回顾会议的书

Topic5 从复杂理论到团队自组织



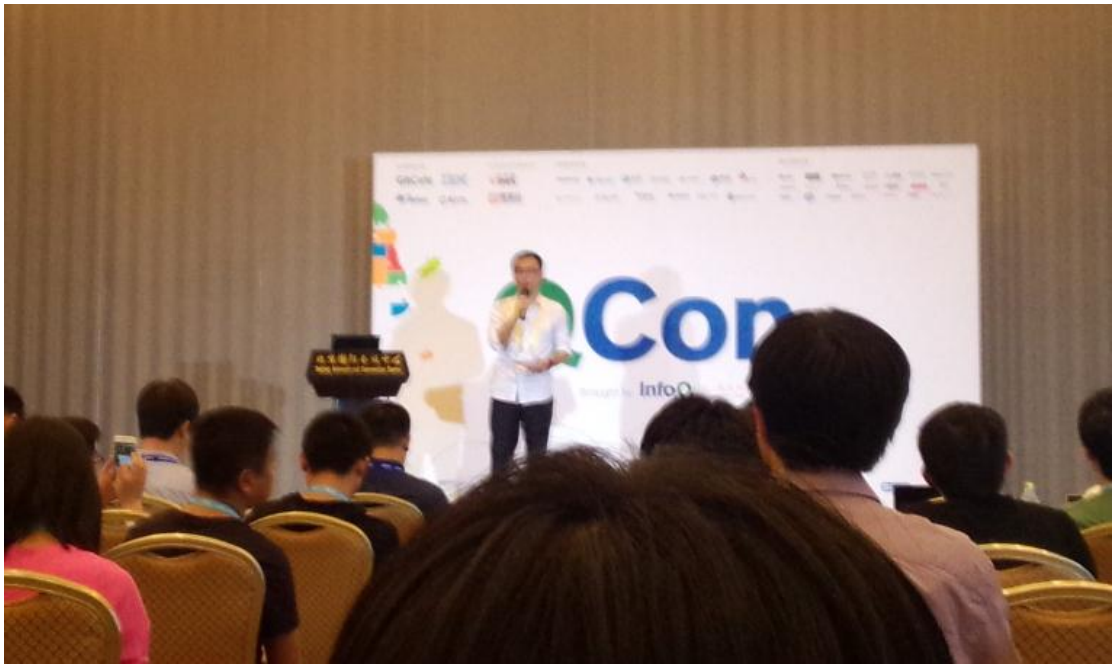
回顾

讲师是吴穹，一个熟悉面孔，时间不够所以都没有讲到团队自组织，主要讲了实施全面敏捷，IT 敏捷了不够，要让业务也敏捷，打广告他们正在做这样的事情，给业务当敏捷顾问，然后介绍一些书《21 世纪的管理挑战》《失控》《复杂》《预知社会》，主要是想说明他们理念都是有理论支持的，思路就是放手，确定适应性指标，快速迭代，适者生存，杂交，突变等。然后重点吐槽 KPI 的无效分解，吐槽不要跟业务提产能，不是你把他骗了，就是他把你给坑了。

反思

我知道讲师分享的有些东西**是来自于我们的 case**，要佩服他们的学习能力和洞见能力，面对复杂的问题能够快速通过学习归纳，然后升华出解决方案。

Topic 功能测试自动化在敏捷过程中的实践与技术展望



回顾：

讲师是大连一家专门做第三方验收测试的公司，主要是讲了他们敏捷实践中测试的两个典型问题，一**是快速迭代以后给测试带来了巨大的压力，二是回归测试压力山大**，解决办法第一主要是测试要 follow 住迭代的节奏，不要在迭代开始的中后期才介入，尽量前置加入到 BA 和 DEV 的沟通中，确保 backlog 中的 story 是通过 BA,DEV,TEST 三方沟通过的。另外是自动化，他们建立行为驱动的自

动化测试体系，通过关键字驱动，设计可执行的用户故事，通过 CI 快速得到反馈从而达到验收驱动开发。看上去美好，其实也有很多坑的，一个是关键字太多不好管理、维护和重用，另外不能并行执行自动化导致自动化执行时间不可接受。最后吹了他们在做的一个事情，就是把这个方案**打包成为一个云服务**专注第三方验收测试。

反思：

其实他们遇到的问题和我们一模一样，他们的解决思路也和我们一样，就测试前置而言他们比我们彻底，我想我们可能还是有组织壁垒或者工作惯性在里面。另外他们这个自动化的测试体系也是和我们实践的基本一样，但是，但是他们确想到了封装融合打包成云服务的想法并且正在实施，真的是佩服他们敏锐的产品化嗅觉，先不谈他们的这个云服务到底能不能成功，但是就**产品化的思路**已经完全超越了我们。

后记

“In short , software is eating the world”，当今正处在一个 IT 的黄金时代，属于软件工程师的舞台很宽广，所以我们要把握好机会，不光是能够实现能够交付，还要敢想、敢干、敢吹。

三天很快就过去了，首先感谢公司提供的这个机会，可以让我直观的了解当前业界一些动态趋势，另外要感谢我们分组的同事他们努

力进行迭代的时候我在轻松听讲座，最后还要感谢北京这几天不错的天气。