

T O P T E C H T E A M

中国顶尖 CHINA

2016年 / 第六季

技术团队

| 访 | 谈 | 录 |



扫一扫，了解更多

Geekbang
极客邦科技

InfoQ

Broadview®
www.broadview.com.cn

FreeWheel CTO：技术创业十年，我从中学到了哪些经验和教训

InfoQ专访FreeWheel CTO 于晶纯女士，咨询了职业发展、选择、团队管理等问题。

专访奇虎360陈斌： 开源与安全并存，阶梯助技术成长

InfoQ采访了360技术委员会主席陈斌，让我们一起来了解开源背后的工作。

专访饿了么张雪峰： 未长夜痛哭者，不足以语人生

InfoQ编辑采访了张雪峰，请他分享饿了么的发展历程，以及团队中一些有趣的独家内幕。

专访符茂松：Twitter新开源的Heron是否可以完全取代Storm

InfoQ在第一时间联系了Heron的技术主管符茂松，了解一些技术内幕。

专访链家网前端总架构师杨永林：8年架构师成长之路

8年的前端开发生涯中，“教主”是如何一步一步地成为知名前端架构师的呢？

中国顶尖技术团队访谈录 第六季

本期主编 郭 蕾

流程编辑 丁晓昀

发行人 霍泰稳

联系我们

提供反馈 feedback@cn.infoq.com

商务合作 sales@cn.infoq.com

内容合作 editors@cn.infoq.com





Mission

使命

整合全球优质学习资源
帮助技术人和企业成长

Vision

愿景

全球领先的技术人
学习和交流平台

企业成长

EGO | EXTRA GEEKS' ORGANIZATION NETWORKS | 高端技术人员学习型社交网络

技术专栏、迷你书、顶级技术大会、垂直技术峰会

Git | GEEKBANG INNOVATIVE TRAINING 极客邦培训 | 一线专家驱动的企业培训服务

众筹小班课、翻转课堂、会员学习计划
技术图谱、技术解惑

技术人成长

会员制、城市分会、学习活动、全球领导力峰会

InfoQ | 专注中高端技术人员的技术媒体

企业内训、工作坊、训练营

StuQ | 实践驱动的IT职业学习和服务平台



卷首语

架构师，让技术流露性格

作者 ArchSummit 大会主编 龙永昕

本期杂志文章的主题都略带深沉，但是这也是符合技术人个性的，在深沉中为了自己的技术人生的呐喊，呼唤那个最精彩的自己。

浪迹技术圈这几年来，我一直认为技术是有性格的。或易或简，或张扬或稳重。隐藏着创立者的意气风发，也关乎那些忠实拥护者们的孜孜以求。

架构师，则是这种性格的代表。他们有太多的曾经，有曾经敢言破除一切推倒重来的豪气，也有曾经搔首而求不撞南墙不回头的韧劲。回首架

构师的成长路，足迹深浅，沉积的是阅历，是深深的智慧。

这让我想起了我们的 ArchSummit 全球架构师峰会。每年 ArchSummit 立项之初，我们都将寻访全求各地的知名架构师。听他们的技术故事，品他们的技术人生，话他们的技术精彩，与他们一起探寻架构师内心对技术的感动。与其说是在探寻如何更好地了解架构师们的所思所想，倒不如说是如何更加了解架构师们的性格。在收获满满的感动的同时他们的建议也给

Preface

ArchSummit 带来了不一样的精彩。

那今年会有哪些不同往昔的精彩呢？今年 ArchSummit 大会上 Twitter、LinkedIn、Cloudera 在《大数据与个性化》专题的争奇斗艳；腾讯 QQ 空间、阿里无线、百度贴吧、58 同城在《发展中的移动架构技术》专题各展乾坤，均得益于此；VR 交互、服务机器人、智能驾驶...《虚拟现实与人工智能》专题新技术各露头角；Uber 实时车辆追踪查询系统演化、滴滴出行架构升级、美团外卖架构演进，《分享经济下的架构探索》

专题各系统架构求索不倦。可以说，ArchSummit 大会上的每一专题都是一次思考，都是一次为更好地理解架构师性格的探索。

同时，ArchSummit 自身的成长也似一个技术人的成长。我们精心呵护，致力于让它某天拥有架构师般的性格，像架构师一样成长，而后自强不息，而后奋斗不止。于此，衷心感谢陪伴 ArchSummit 成长的架构师们，也衷心期望 ArchSummit 能成为你们的知己。



EGO EXTRA GEEKS' ORGANIZATION NETWORKS

我们的使命：
引导高端技术人学习和成长

我们的愿景：
全球最具影响力的高端技术人社交网络

关于EGO:

EGO是极客邦科技旗下高端技术人聚集和交流的组织，旨在打造全球最具影响力的高端技术人学习和成长平台，线上线下相结合，为会员提供专享服务。

EGO采用实名付费会员制，每一位申请加入的会员都必须经过EGO组织的严格审核，保证会员信息的真实性，让每一位会员在平等和相互信任的环境中分享交流，大家共同学习，共同成长。



微信公众号



即刻加入EGO

为什么 加入EGO?

学习交流

独一无二的学习和交流体验，会员间平等分享各自的实战经验，从工作到生活

拓展人脉

认识国内外志同道合的技术同行，分享交流，拓展自己的人脉圈子

获取信息

获取最新最重要的行业信息及资讯，了解最新技术动态、新闻、社会热点，独家原创的专家观点及评论

解决问题

身边的智囊团，遇到问题时能够第一时间通过EGO平台找到答案

工作之外

在日益激烈的竞争中平衡工作与生活之间的关系，提高软技能

入会资格

1. 认同EGO理念，遵守EGO规则；
2. 热爱学习，乐于分享，追求成长，相信技术的力量
3. 公司/团队中的技术决策者，如CTO、首席架构师、技术总监等；
4. 在某一技术领域具有超过5年的工作经验。

EGO客服微信号: egowinner EGO客服邮箱: service@egonetworks.org

了解更多信息请访问EGO网站: <http://www.egonetworks.org>

FreeWheel CTO : 技术创业十年，我从中学到了哪些经验和教训



作者 郭蕾

于晶纯，现任 FreeWheel 公司联合创始人兼首席技术官，引领着 FreeWheel 的技术方向。于女士拥有近 20 年的广告行业和技术管理经验，她组建并发展了一支分布在北京和纽约两地、规模近 300 人的 FreeWheel 全球研发团队，全面负责团队运营以及产品的研发和运维工作，并带领团队不断进行产品架构的改进和技术创新。

FreeWheel 是美国领先的视频广告管理和投放平台，目前在北京和纽约两地设有研发中心，有近 300 人的研发团队。其联合创始人兼 CTO 于晶纯女士（Diane Yu）有近 20 年的技术管理经验，从 DoubleClick 到自己创业，能在一家美国公司中从程序员做到 CTO，这样的华人

女性并不多见。InfoQ 就于晶纯女士的职业发展、选择、团队管理等问题对她进行了专访。

InfoQ：在创立 FreeWheel 之前，你曾在 DoubleClick 工作。而在 Google 收购 DoubleClick 的时候，你却选择了离职。能介绍下这段经历吗？

Diane Yu: DoubleClick 是非常好的平台，在那里工作的九年于我来说非常珍贵。那时候基本上每两年我的工作内容都会有很大的不同，包括工作范围、职责等等，我学到了很多。回顾在 DoubleClick 的最后两年，我感触比较深的一点，是一支齐心协力做事的团队能够释放难以想象的巨大能量，相应地，做成事情的几率也很大。那么我呢，一直以来有一个梦想，就是回到中国，创建一支我自己的团队，按照我的想法塑造它的文化和发展。我认为一定能在在中国建立一支最棒的工程师团队，齐心协力地做事情。当时摆在我面前的这个机遇，能够帮助我实现这个梦想，所以我决定试一下。

当时我的两个合伙人想做一个视频货币化平台。他们一个有很多很好的想法，另一个有很好的销售技能，就缺一个能够把想法实现的人。机缘巧合，他们找到了我。我本身对他们的想法很认同，同时我也提出了两个条件。一个是我要用我的方式和产品团队合作，另一个就是回到中国，按照我的想法建一支工程师团队。所以 FreeWheel 从建立的第一天开始，就秉承着这样一个宗旨——在中国建立一支全世界最棒的工程师团队。

我曾经带过很多来自不同国家不同背景的工程师团队，我觉得中国的程序员是

被严重低估的。中国的程序员有非常好的特质，善于学习，非常勤奋，这本身就很难得。更何况北京有这么强大的人才储备，每年有那么多优秀的毕业生选择留在北京。这也是吸引我回到北京的很大原因。

InfoQ: 国内很多人说女性不适合做编程相关的工作，因为到一定程度，会有天花板。而你却在一家美国的公司里做到了 CTO，非常成功。你认为突破这些天花板需要哪些品质，比如说勤奋、坚持、正确的选择？还是什么？

Diane Yu: 我其实认为女性挺适合做这行的。对于编程而言，很多时候决定一个程序成败的关键可能是一行非常简单的代码，而女性独有的细致和直觉在工作中常常给我很多帮助。

女性会不会遇到天花板呢？肯定会。在我的职业生涯当中，随时都会碰到。举个简单的例子来说，在参加会议的时候，我经常会发现周围的人全是男性，而且级别越高越是如此。在这样的情况下，我觉得职业发展的首要素养就是积极做事的心态。尤其在发展快速的 IT 公司，职责界定有时候不见得非常明确，这恰恰是你展示自己能力的时候。

用积极的心态面对并解决问题，而不是等着老板来告诉你做什么，你将会在很短的时间内引起老板的注意并被重用。其

实真的就是这么一点点区别。同样面对一件事，有的人看到的是让人头疼的问题，有的人看到的则是契机，一个可以通过解决问题来学习新知识和新技能的机会。我的职业生涯其实很多时候正是得益于这种态度，比方说我的第一个职业里程碑。

1998年一家非常大的搜索引擎公司跟DoubleClick谈合作，当时这个合作机会可以给DoubleClick带来一笔相当可观的收益。但是这家公司列出了一个签合同的必要条件，就是我们要能提供关键词词条的预测功能。当时DoubleClick的产品没有这个功能，并且团队作了相应评估之后也认为没办法在规定的期限内做出这个功能，基本上已经放弃这个合作了。这时候，销售团队的Mehdi直接找到了还是一名程序员的我，问我愿不愿意试一试。

我想反正也不会失去什么，也就没什么压力，就答应了。那是我第一次挑战自我能力的极限，跟我合作的QA也是白天黑夜跟我一起工作。两个星期之后，这个功能测试成功了，随时可以上线。在庆功会上，Mehdi走过来给了我一个非常大的拥抱，说没有我就没有这次签约。这个事件给了我一个非常好的展示自己的机会。我也很幸运，试成了。但是如果我当初没有积极的心态，把做这个事看成负担，那后面的一切也就不会发生了。

第二个我觉得同样很根本、很重要的一点，借用Steve Jobs说过的一句话“Stay hungry. Stay foolish.”能够从每天的所见、所闻当中不断发现可以学习的东西，像海绵一样吸取这些知识，你一定会比别人走得更远。说起来轻松，但是真正能做到这个的人不多。我喜欢接触新技术，倾听新想法，跟各种各样的人一起合作，只要能（从周围的人身上）学到东西，我就会觉得很享受，很快乐，做事情也不觉得辛苦。拥有这样一种好奇心，你就会看到的比别人多，学到的比别人多，生活也就充满惊喜。

前面提到的两点是我的一点心得体会，跟性别关系不大。不过实事求是地讲，女性看问题的角度和与生俱来的性格特点跟男性确实很不一样。比如男性更善于自信地表达自己的想法，乐于成为关注的焦点，而很多女性不是这个样子，往往不敢去表达或者坚持自己的观点，所以职场里女性的声音往往很弱。这其实会成为女性职业发展的一个障碍。我就面对过很多质疑的目光，尤其在这个男性占多数的职业里。很多人一开始可能会觉得你一个女生真的懂吗？我的做法是忽略性别差异，忽略这些质疑的目光，把注意力集中在事情本身，这能帮助我自信地表达观点，也帮助我最终赢得大家的尊重。

InfoQ：你在 DoubleClick 工作了 9 年，我想这 9 年时间你一定成长了很多，这些经历也为你之后创业的成功打下了坚实的基础。可以展开谈谈吗？

Diane Yu：这个问题让我想到了 FreeWheel 做校园招聘的时候，很多大学生问，我毕业后是去创业好还是先工作好？从我个人的职业经验出发，我觉得要有一定的经验积累，再去创业，成功的机率会高一点。因为你积累的这些经验在你以后的创业道路上都会发挥作用。创业和给别人打工是完全不一样的，如果什么积累都没有，单凭一腔热情投入到创业大军里，有可能被碾压得粉身碎骨。

创立 FreeWheel 之前的积累对于 FreeWheel 的今天，在一定程度上起了非常关键的作用。2000 年互联网泡沫，我人生经历了第一次极为痛苦的管理经历。当时公司的绝大多数客户在几个月的时间里全部关门，可以想象这对公司营收是一种什么样的打击，公司股票也直线下跌，怎么办？全线裁员。当时所有人都做好准备，一轮一轮地裁员，直到最后一个，然后关门。这次经历确实非常痛苦，但是在这个过程中，我学到很多东西。

第一，学会了如何生存。如果不能生存下去，即使你的产品再好、团队再好、想法再好，你只能关门，什么也没有机会

实现。

第二，不能盲目扩张。经营一家企业最主要的是资金流不能断，否则一切难以进行。我们三个人（注：FreeWheel 的三人创业团队）都有过类似的经历，所以对我们来讲创业这么多年，我们有了很棒的团队，不断地收到客户订单，但是这根弦从来没有松下——如果银行账户流动资金不足，工资发不出来，公司就随时可能关门。

第三，如何用最小的损失保全团队的战斗力。在痛苦和泪水中，我学会了如何在逆境中去鼓舞团队士气，因为一个没有战斗力的团队就等于零。最痛苦的时候，裁员大家都可以理解，但是大家最担心的是你能不能做到公正透明。在 FreeWheel 的发展过程当中，扪心自问，我尽了最大努力去做这一点，对人和事做公正的评价。我们所有的升职评估，不管多困难，大家都会坐在一起进行讨论，坚持公正透明，要知道这是需要很多精力的。正因为我们做到了，公司留住了很多人才，我们也有很多优秀的老员工从 FreeWheel 成立到现在一路跟我们走过来。

InfoQ：聊聊你在技术管理方面的经验？你是如何理解扁平化的组织架构的？

Diane Yu：作为一家技术公司，特别重要的一点就是技术线跟管理线要平等

对待，我觉得这个很重要。很多公司容易偏重管理线，对技术线的重视程度不高，导致技术线缺乏清晰的职业发展路径。当一个公司的管理线跟技术线不真正平等的时候，技术的声音就会越来越弱，会导致公司在资源分配和考核时给技术的权重不够。时间久了，你会看到公司的技术平台就会慢慢过时，不再有自我创新的动力。

FreeWheel 非常重视这一点，我们的技术人员就有非常清晰的职业规划。举个例子来讲，FreeWheel 的总架构师和高级副总裁是平级的，如果你钟情于技术，并不需要一定成为管理者，走技术的路线一样可以得到职位和薪水的晋升，像架构师、首席架构师这样的技术职位，在 FreeWheel 都有。我们能够做到技术人员的薪水、级别以及声量跟管理岗的人员平级。比如，因为总架构师和高级副总裁平级，在管理层会议的时候，总架构师就有比较大的话语权，尤其是偏技术的问题。管理者在决策的时候就不会是管理岗一边倒的决策，而是一个管理和技术平衡的决策。

我对于扁平化管理的理解是，这种模式的关键在于“Open-door Policy”（注：开放式管理）。FreeWheel 有硅谷的 DNA，我们非常赞同扁平化的管理理念。所有人可以随时找我，我不认为自己是

老板，当有人喊我“于总”的时候我也会觉得有些不习惯。这也是开放式管理的一个方面，就是上下级的界定没有那么清晰。这样做有非常大的好处，就是你可以听到很多不同的声音。我不怕员工跟我拍桌子，只要他 / 她的意见中肯、有可行性，我愿意看到员工坚持自己的意见，而不是老板说什么都点头称道。这样能够说明大家对于自己专注的领域是有热情的。我个人就很鼓励员工勇敢地说出自己的想法。

你今天跟我拍桌子，明天照样可以升职。我觉得这是一个良性循环，员工越勇于表达自己，管理层越能够听到更多不同的意见，也就更便于做出正确的决策。在这样不断自我修正和自我激励的环境中，去除一层一层的管理层级，让决策者真正听到一线员工的声音，公司才能健康地发展下去。

InfoQ：你帮助 FreeWheel 把研发团队从 0 扩张到了 300 人，而且还是一个跨国的工程师团队，这其中一定有很多的经验教训。如果说有三个关键的点，你认为是什么？

Diane Yu：这样的经验教训太多了。如果只说三点的话，第一就是一定要招最优秀的人才，而且要留住他 / 她们。这一点说起来很容易，但实际做起来特别难。因为好的人有很多机会，你需要给他 / 她

一个留下来的理由，你需要有能力激发出他 / 她的工作热情。如果不这样做的话，一段时间之后，你就会发现优秀的人才都走了，那么公司就会趋于平庸，危机也会随之而来了。

对于如何界定人才是否优秀，我有一个比较严格的标准——跟我一起工作的人，一定要在某些方面比我强，让我能够怀着敬意跟他们一起工作。当你每天跟你非常敬佩的人一起工作的时候，这个工作就一点都不辛苦，因为每天都能够不同的人身上学到新的东西，这是非常令人振奋的事情。

另外一个很关键的点，尤其是对创业公司来说，就是在意识到某个错误之后，能够及时纠正。这对于整个团队的发展至关重要。首先，团队的其他人会发现这个错误，如果你知道错误而不去纠正，对于其他非常出色的人是不公平的。其次，很多时候这种错误不在个人，而在领导。比如把对的人放在了错误的位置上。这样的问题不纠正，这些人才的青春就白白浪费了。所以从我以往的经验来看，在第一时间纠正错误，无论是对公司还是个人，都是非常负责的。

最后，我认为很重要的一点，是把握住最关键的问题，然后尽一切努力克服随之而来的困难。鱼和熊掌不可兼得，没有

任何一个解决方案是完美的，但是只要你抓住了最核心、最关键的部分，就可以了。FreeWheel 作为一家跨国公司，从最初公司成立，就决定在中国建立核心研发团队，这个对于我们来说是核心问题。那么我们在做出这个选择的同时，一定要勇于面对并克服随之而来的其他问题，比如沟通和文化壁垒、商业模式和客户需求方面的壁垒等等，这些都是客观存在的。正因为我们认为，招到最优秀的人才是我们最看重的，所以我们采取了很多措施来应对上面提到的这些问题。比方说我们提供英文培训和海外轮岗机制，让程序员有机会和客户面对面，去了解客户的痛点在哪里；管理团队和产品团队会经常来北京，跟我们的团队面对面讨论问题，等等。

InfoQ：你如何看待时下热门的微服务架构？

Diane Yu：最近我们团队内部也在讨论微服务架构。微服务架构有一个好处，就是承诺了一个愿景，考虑围绕业务领域组建来创建应用，这些应用可独立进行开发、管理和迭代，从而使管理和服务功能的交付变得更加容易。但是我个人认为，这个概念很简单，但要把这一点做好并不容易。微服务架构特性决定了每个微服务实现的功能要相对独立，同时不能太复杂，复杂的应用要多个微服务协调产生。那么

如何在保持微服务敏捷特性的同时，高效地组建复杂应用，就是最关键的问题。尤其是对于 FreeWheel 这类业务领域相对复杂的公司来说，客户都是大企业，应用很复杂，这其实对架构师的水平是更高的挑战。

我认为微服务架构最近之所以很火的原因，是因为它针对业界普遍存在的一个技术痛点，提供了一个“看上去很美”的解决方案。但是，所有的技术方案都不是普适的，选择任何一种解决方案都有它的优势及不可避免的劣势，对于企业来说，是不是适用微服务架构其实在于你能不能掌控它的劣势，并充分发挥它的优势。我们目前在一些业务领域讨论实现微服务架构，究竟是不是一个真正行之有效的手段可以解决我们的痛点，我认为尚需一些时日假以证明。

InfoQ：最后一个问题，您认为一个好的程序员，应该具备哪些优秀的品质？

Diane Yu：除了我之前说的一些之外，我认为一个好的程序员要有想法。有想法是什么意思呢？一般来说，程序员需要和产品团队紧密合作，而一个普通的程序员和一个出色的程序员的区别就在于，出色的程序员会成为产品的合作伙伴，普通的程序员则只能成为产品的工作人员。而合作伙伴的概念就是，当产品的负责人

遇到问题的时候，会第一时间找你商量解决方案，而不是产品布置一项工作给你执行这么简单。

好的程序员要具备产品方面的想法，和产品建立互相信任的关系，能够在产品遇到问题的时候提供行之有效的解决方案。在这个过程当中，就要求程序员不要把自己限制在自己的领域，不能想着“我是搞技术的，我不管产品”，或者“我是编软件的，运营的事情与我无关”。要能够看到更大的格局，能够站在别人的角度去思考问题。比方说你在写软件的时候如果能考虑到这个产品如何运营、如何销售，如果你能够把这些东西考虑得很好的话，你会发现你就不仅仅是一个写代码的“码农”了。

专访奇虎360陈斌： 开源与安全并存，阶梯助技术成长



作者 徐川

陈斌，奇虎 360 技术委员会主席，360 开源委员会成员，Web 平台部总监的服务端技术专家。

参加开源社区的活动，将自己的代码开源，为知名的开源框架做出自己的贡献，这对于很多技术人员来说，都是提高技术水平的阶梯。然而，很多公司并不支持技术人员这样做，他们担心开源会给自己的产品带来安全问题，技术人员也因此丧失了很多学习机会。那么，开源真的不能与

安全并存吗？

奇虎 360（下文简称“360”）作为一家以安全平台为支撑的公司，在企业内部建立了一套开源体系，不仅推出了优秀的企业级开源框架，还帮助了技术人员成长和学习，得到了很多意想不到的收获。奇虎 360 是如何做到这些的呢？为此，

InfoQ 采访了 360 技术委员会主席，360 开源委员会成员，Web 平台部总监的服务端技术专家陈斌老师，让我们一起来了解开源背后的工作。

InfoQ：360 目前开源现状是什么样的？有哪些代表性的开源项目？

陈斌：360 的开源现状是由下至上形成的。三年前，360 的一些小团队就开始做了开源分享，去年，我们整合了 360 内部的 GitHub 账号，设置了主账号，开始着手进行统一的管理。

目前，360 在 GitHub 的开源项目约为 20 个。其中，比较有代表性的有三个。一是被成为安卓黑科技的 DroidPlugin，它可以在 Android 上免安装、免修改来运行一个 APK，是一款比较成熟的插件化开发框架。还有一个比较成熟的 MVC 框架——ThinkJS 它是一款使用 ES2015/2016 特性全新开发的 Node.js MVC 框架。三是 Atlas，它是一个基于 MySQL 协议的数据中间层项目。这三个开源项目影响都比较大，其中，DroidPlugin 目前已有三千多个 star。

InfoQ：360 有参与过 OpenResty 这样的国际型的开源项目，时是否有相关制度？是否 patch 或 fork 过大型的开源项目？

陈斌：对于国际的开源项目，我们主

要是关注和使用。获取到开源项目后，我们会进行使用，根据具体情况做一些修改，遇到 bug 时会提交给官方。这方面没有统一的制度去参与

像 Redis、OpenResty，包括一些 iOS 的相关项目，我们都有提交过一些 patch。由于我们是做安全行业的，所以在使用开源项目时，会做一些特殊规定，所以会 fork 一些分支出来。

InfoQ：360 的开源理念是什么样的？

陈斌：我们是鼓励开源的，并且认为开源会更有利于项目。首先，我们不担心核心技术被别人学习。现在互联网如此发达，核心技术早已不是最重要的核心，而快速提高技术水平才能保持领先，才是最关键的核心。开源是一种趋势，也是提高自身水平的一种手段，因此，我们认为，该开源技术的都应该开源，而且要给予正确的导向，鼓励开源的同时保证做到安全地开源。

InfoQ：请介绍一下 360 开源委员会，主要成员以及运作模式等。

陈斌：360 开源委员会是今年成立的，主要包括四个部门：技术分委会、安全审查部门、法务和市场。

每个部门都有其专门的职责：技术分委会的主要成员是架构师和工程师，负责

鉴定项目用到的技术是否适合做开源；安全审查主要是对开源项目的安全性进行审查，防止泄露公司的账号等对安全造成威胁的信息；法务部门会审核是否符合我们的开源协议；市场部门负责对开源项目做推广工作，例如做开源大会等。

InfoQ：360 内部的开源流程是怎么样的，大概周期需要多久？

陈斌：360 内部的项目想要开源有规范的流程的。

大致步骤如下所述。

- 技术审查。要经过技术委员会的审查。他们会分析其中的技术要点以及用户场景，来评判项目是否适宜开源。
- 安全审核。由安全部门对源代码进行审核，确保不会对敏感信息造成威胁。
- 法务审查。由法务审查是否遵循相关协议，例如BSD协议。
- 开放代码。在GitHub上建立仓库，提交开源代码了。
- 推广使用。接下来，我们还会推动开源项目的使用，不仅推动内部使用，还会由市场部来做相关的推广。
- 长期维护。一个好的开源项目，必须有人对其进行维护，我们会

将项目绑定到团队身上，由团队来负责长期维护。

一个项目由提交，经过各种审查，到最后开源至 GitHub 上，大约需要两周。这些过程都是在工作时间完成的，因为 360 不仅仅是一家安全服务公司，也是一家技术公司。

InfoQ：刚才提到对开源项目进行维护，那么 360 如何保证公司级开源项目的持续维护呢？

陈斌：公司级的开源项目需要投入源源不断的人力，而单个的人是具有不稳定性的，假如将它绑定在个人身上，是要冒着很大风险的。所以我们将开源项目和团队绑定在一起，由固定的团队来负责某一开源项目。这样一来，实现持续维护就容易多了。同时，由于项目的持续维护有了保障，口碑必然增强，不管是个人还是团队，荣誉感都会大大提升。

InfoQ：请介绍一下 360 技术委员会，它的主要职责是什么？

陈斌：360 技术委员会是我们的一个虚拟组织，对应各个领域（如前端）设立相应的分委会。具体的成员是由技术人员推选出来的，比例为 5%，即 100 位技术人员中会推选出 5 位委员。

技术委员会的职责主要有以下三方面。

技术评级。360 内部建立了评级体系，帮助技术人员成长。很多程序员都遇到过这样的问题：刚入职时被分配到某个项目中去，这个项目由于各方面的原因做的并不成功，而后续的发展却要与该项目挂钩。这对程序员来说是非常不公平的，因为一个项目的好坏并不能代表程序员技术的高低。我们给技术人员建立了一个成长的阶梯，不以项目论成败。而 360 技术委员会在这个体系中担负着重要的职责，就是为技术人员进行评级。

内部沟通。由于 360 内部的管理比较扁平化，所以会带来很多沟通上的问题。360 技术委员会会来改善这种情况，打通各个团队之间的交流通道，横向沟通，学习各个团队的经验，避免重复“踩坑”。

对外宣传。360 技术委员会还肩负着对外宣传的职责。我们每个月的第 3 周的周六都会有对外交流的活动，而且还有 360 训练营，所有技术人员都可以免费参加这些培训活动。

InfoQ：360 内部有一些非常优秀的技术团队，如何保持它们的成长？

陈斌：保持团队的成长就要保证对技术人员有正确的发展导向。首先，我们将薪资与技术级别挂钩，从薪资上直接激励技术人员提高自身技术水平。其次，我们还会为优秀的技术人员提供学习机会和环

境，例如参与技术大会等。另外，我们还设有一系列的荣誉，例如对高级别的技术委员会委员颁发奖杯。

InfoQ：360 内部的团队非常扁平化，能说说它所带来的好处和坏处吗？

陈斌：扁平化的管理的一大好处就是可以更好地适应变化，因为扁平化的团队更加灵活，能够快速解决问题、快速发展。而与此同时，却带来了沟通减少、继承性差、难以积累所学等问题。总体来讲，其优势还是大于劣势的。我们正从各个方面进行努力，来消减扁平化管理所带来的坏处。

InfoQ：扁平化的管理会造成公司业务由众多的小团队去分别完成，那么如何避免他们踩同样的“坑”？

陈斌：我们有专门的技术部门，例如 Web 平台部，它们不直接和业务挂钩，有自己的 KPI 和工作职责，来做技术方面的总结和升华。这样可以避免小团队难以积累的缺点。

InfoQ：公司内的开源项目如何在不同团队进行推广？

陈斌：推广的方式有很多，除了进行宣传外，我们还会在评级时对使用了 360 开源项目的人员进行加分，这可以直接促使他们使用公司内的开源项目。

专访饿了么张雪峰： 未长夜痛哭者，不足以语人生



作者 陈兴璐

张雪峰，饿了么 CTO EGO 会员；70 后，目前带领饿了么技术团队。之前带领携程软件架构团队 & 框架研发团队，后担任携程国际 BU CTO。曾有过一次创业经历（教育行业），深知创业之痛并快乐着的感觉，理解创业之苦、之难、之惨烈。

潘石屹说，现在是创业的黄金时代。饿了么作为中国互联网创业圈大只独角兽，2014 年底研发团队大约 35 人，2015 年春节后翻番到了 70 人左右，2016 年春节后上海、北京两个研发中心达到近 900 人，预估 2016 年中接近 1000 人。在此过程中，随着业务极速发展，组织架构变化、研发流程优化更是家常便

饭，个中酸甜苦辣。不过无论如何变化，有一点却是不变的：研发管理，没有所谓 Best Practice（最佳实践），只有适应业务发展且生命周期有限的 Suitable Practice。

InfoQ 编辑采访了张雪峰老师，请他分享这一路历程，以及饿了么团队一些有趣的独家内幕。

InfoQ：在饿了么技术团队中，你能描述一下你作为 CTO 的角色职责吗？

张雪峰：主要是五方面职责：

- 找到合适人才；
- 组建合理团队；
- 稳定交付产品；
- 极致经营产品；

当然，个人认为最重要也是最难做到一点：如何让团队 continuously enjoy working and result？如果无法让团队始终充满激情，不可能有持续极致成果。

InfoQ：请介绍一下你离开携程加入饿了么的初衷，饿了么最吸引你的地方在哪里？

张雪峰：我外表看着文静或行事貌似稳重，但骨子里还是一个比较喜欢折腾的技术男。在携程稳定工作两年多后，出现了三个机会：

1. 携程鼓励内部创业，当时和几个志同道合同事准备申请内部创业项目，主要涉及旅行规划与全流程服务（大概是蚂蜂窝+ 世界邦 + 妙计旅行合体，不同点是可基于携程已有丰富数据积累），不过有点可惜，在集团层面汇报 BP 时被毙掉了；
2. 当时计划和几个朋友出来创业，做物流相关产品（不同于饿了

么目前重点发展的 [即时配送]）；

3. 接受当时朋友Mark（张旭豪，饿了么 CEO，四大饿人之首）、Raymond（汪渊，饿了么联合创始人，四大饿人之一）邀请，加入饿了么。

加入饿了么之前，我主要作为 Raymond 私人技术顾问，偶尔出出主意或聊聊技术。后来，Mark、Raymond 和我谈了几次，了解到我心有纠结后，直接告诉我，也别考虑什么大公司内部创业或和朋友出来创业了，直接加入饿了么，这两问题不都解决了吗？后来仔细想想，也真是这么回事，那就直接加入吧。

饿了么最吸引我的主要原因不是 big title，而是创始团队尤其 Mark & Raymond 给予的 fully trust。考虑加入一个新团队，最高层信任比其他什么都重要。另外一个原因就是年轻有冲劲，一个以 90 后为主的创业团队，感觉和我在携程时的 80 后为主乃至平级以 70 后居多的团队，感受完全不一样。

饿了么除了王祖蓝那句传播较广的“饿了别叫妈，叫饿了么！”，还有一句内部流传更广且被很多同学视为行动座右铭的话：饿了么，和你一起拼！

InfoQ：2015 年你入职饿了么的时

候，研发团队 100 人左右，而现在整个队伍已经超过 1000 人，你是如何看待这样的极速发展？

张雪峰：人数 10x 不难，主要难在三点：

1. 系统稳定性是否可以同步 10x 提升？
2. 需求同步 10x 爆炸后，是否可以有所为有所不为？
3. 个人认为最重要也是最难做到一点：组织架构是否可以持续敏捷并适应软件架构的不断变化与调整？

关于第一点“系统稳定性”，主要涉及四方面建设：流程规范、理念提升、技术保障、高效工具。

关于流程规范，人少可以吼一嗓子马上解决问题，人多肯定不行，尤其人多了大家可能不在一个楼层甚至不在一个城市办公，流程规范虽有所制约，但必不可少。

关于理念提升，和技术文化还不太一样，很多时候，一起故障引发客人投诉或业务吐槽，或长时间无法恢复导致严重事故及业务损失，比不断强调系统稳定性或生产无小事更有效，很多时候，反面案例或身临其境都是最好的“理念提升”教材。

关于技术保障，在绝大部分大型互联网企业中都是核心技术团队，只不过，绝

大部分时间他们都以 Silent Hero 方式坚守一线，为所有其他技术团队保驾护航。

在当前开源理念盛行、分享无处不在大环境下，高效工具（Engineering Productivity）不是没有，而是太多，无论拿来主义还是自研，关键看是否适应现有团队习惯（人少可以强扭，人多只能适应）。工具就像春晚，人人满意基本不可能，能满足大部分同学核心诉求即可（少数服从多数）。

关于第二点“有所为有所不为”，主要还是取舍问题，PM 需要取舍，Arch 同样需要取舍。在团队爆炸、需求爆炸的极速发展阶段，对 PM 或 Arch 来说，绝大部分时间在做的事，就是“两害相权取其轻”。

关于第三点“组织架构和软件架构间关系”，康威定律已经讲得很清楚，如果说有点新意，无非就是针对不同类型团队的 virtual/override。

InfoQ：在饿了么工作一年的时间里，如何带领整个技术团队？遇到过什么样的困难？经历了哪些重大事件的考验？

张雪峰：（1）刚才已经提到，基本就是按这个思路和时序，不断循环优化；

（2）大困难主要两方面：系统稳定性 10x 提升，团队稳定性尤其中坚力量稳定性尽可能保持不变或在可控范围内波动；

(3) 过去一年经历三次 P0 故障：红包金额发放异常、支付完成状态异常、机房核心主备全挂，这三次最严重级别 P0 故障，除了很对不起业务团队及 PR 同学，也让技术团队充分意识到 Design for Failure 重要性与必要性，总结起来就是：一障抵千言，没经历过重大故障处理或造成重大故障的技术人员，很难真正成长起来或独当一面（少数天才或天赋异禀者除外）。

InfoQ：饿了么如何成长为大只独角兽？在与美团的战争中，如何做才能够胜出，或者说不败？

张雪峰：外卖 O2O 的市场很庞大，而且还在不断培育和催熟中。长远来看，高度专注的公司才能持续提供高效率高标准的服务，最终获得用户和市场。饿了么团队从 2008 年开始创业，2009 年正式上线，到现在八年左右的时间，我们一直做着同样一件事情。

目前，围绕“美好生活，触手可得”，饿了么已初步建立起由交易平台、即时配送和供应链组成的到家服务生态体系，通过对整个外卖链条的上下游进行覆盖，从根本上提升用户的体验，并逐步构建起自己的核心竞争力。

InfoQ：听说饿了么的 CEO 很强硬，且从未给别人打过工，你作为 70 后的

CTO，如何看待近 90 后的 CEO？平时工作中会有冲突吗，如何处理？

张雪峰：纠正一下，Mark（张旭豪，饿了么 CEO）不是强硬，是霸气，还不是侧漏那种。Mark 85 年出生，我 76 年出生，但因为前面提到的 fully trust（霸气和强硬区别之一），所以我俩间的配合，不存在（工作）代沟。

平时工作中肯定存在分歧，但没出现过大的异议或原则性冲突。我俩都是理工男，分歧处理方式也挺简单，要么他说服我，要么我说服他，印象中没出现过需要引入第三者协调场景。

InfoQ：你自己曾经创业，对想要创业的人有什么样的创业指导，有什么感悟或经验可以和大家分享？

张雪峰：我那段创业经历虽然辛苦，但需要我管的非技术或非研发杂事不多，这些都是 CEO、人事、财务、行政扛着，我就搞好前面中提到的几点即可，总结起来就是技术人最喜欢的创业五字真言：痛并快乐着。

站在一个技术从业人员角度，结合我曾经创业失败的体会，大概有这么几点分享：

1. 找到真正需求，而不是创造“伪需求”；
2. 先做出最小可用产品（MVP），

然后根据市场反馈再定是继续迭代优化，还是迅速调整方向；

3. 既然很多创业者只缺一个程序员，那就对你的程序员好一点，再好一点。刚开始创业，更需要 full-stack engineer，而不是各种 big title（研发总监、架构师、CTO）；
4. 即使 full-stack engineer，也不是真的全能选手，所以，机房

/网络/服务器/存储/监控/告警/安全甚至数据库什么的，可以考虑适当交给云计算。

我肯定没资格提什么创业指导，强烈建议联系采访上海滩乃至中国互联网创业圈“奇葩”一般的存在：以 CEO 张旭豪为代表的饿了么创始团队，江湖人称“四大饿人”。

关注微信号回复“十年”

看云计算风云变迁



蘑菇街曾宪杰： 技术团队要在**专业上PK**，但不搞哲学讨论



作者 杨赛

曾宪杰，2007 年加入淘宝平台架构组，花名顶天，主要设计实现了淘宝自研的 Notify 消息系统，并参与淘宝其他中间件的相关工作，2010 年下半年负责整个淘宝 Java 中间件团队，将团队打造成业内知名 Java 技术团队。2015 年初正式加入蘑菇街，负责整体技术工作。对新技术有浓厚的兴趣，虽然做技术管理的工作，还是保留了一颗码农的心。著有《大型网站系统与 Java 中间件实践》一书。

蘑菇街近来的发展十分迅猛，这支高歌猛进的团队引起了我们的兴趣，近日我们采访了蘑菇街技术研发部副总裁曾宪杰同学。

InfoQ：能否介绍一下您最近重点在做的事情，以及关注的领域话题？

曾宪杰：最近重点的事情规划美丽说、

蘑菇街、淘世界三家公司融合后的整体集团的技术路线，也在做集团的技术架构的调整和融合。此外，的一个重点是和团队投入在通过混合云来改进现有技术架构。比较关注的是和大数据还有云计算相关的话题。

InfoQ：2004 年底的时候您在重庆

跟人合伙创业过，2007 年加入淘宝的时候淘宝是一个刚脱离创业阶段的团队，去年加入的蘑菇街也是一个刚脱离创业阶段的中型公司。这三次工作环境上的变更，您的新角色都不一样，在进入这些新角色的过程中，哪些事情是您最享受的？哪些事情又是最让您感到压力和挑战的？

曾宪杰：2004 年底离开上海去重庆创业，那个时候也是自己本科刚毕业两年半，还依稀记得当时是兴奋，以前大学看过一本《清华园中的创业启蒙》，还是很想体验下创业的，所以创业本身就让我很享受。2007 年本科同学内推面试淘宝，当时有两个事情自己有压力，一个是面试，当时不确定是不是能通过（毕竟之前的两年半自己在创业中做的事情很杂），另外一个试用期快转正的时候让我转 Java，在淘宝的挑战还是比较多的，毕竟在我在淘宝的前几年赶上了淘宝业务高速发展的时期，对技术的挑战很大，在淘宝最享受的也是这几年的挑战时光。在蘑菇街，自己要为公司整体的技术发展负责，这一年多来确实很辛苦，但是很爽。过程中也会遇到很多困难和问题，但更多的感受是自己肩上的责任。

InfoQ：从您个人的角度，如何看蘑菇街在电商领域的发展空间？

曾宪杰：蘑菇街在很多人眼中是一家

垂直电商，其实在我看来蘑菇街并不是一家仅仅做电商的公司。三家公司融合后的新的集团的使命是“让一半人类更幸福”。我们会充分发挥科技力量，在服务女性的各个领域提供极致体验，那么目前我们主要提供的服务是时尚电商以及时尚资讯。那么单就电商讲，纯货架式的电商已经很难做了，我们希望走的路是电商、社区、内容相结合的方式，给到我们消费者更好的体验。而且在无线时代，我认为我们更要做的事情是围绕人群来做，该做深的地方是要做深，不能仅仅只提供一个很薄的平台。确实会有一些脏活累活，但是首要的是保障我们用户和消费者的体验。

InfoQ：今天的蘑菇街，相比今天的淘宝，以及 2007 年的淘宝，对技术人员的要求有哪些异同？

曾宪杰：互联网包括电商经过了多年的发展，已经有非常多有经验的从业人员了，相比于 2007 年的淘宝，在人员的技能方面的要求会更高，而对于从业经验的要求也会高一些。

InfoQ：最近看了什么有意思的书籍或者文章、演讲跟大家分享？

曾宪杰：最近没有很系统的去读书，基本上是在看 4 月份 QCon 北京会议的资料，还有 InfoQ 架构师的资料，更多的时间还是跟团队一起在看如何解决我们面临

的具体的技术问题。这些也是多年来的习惯。后面如果有时间，我自己会想看一些跟机器学习相关的书，自己之前主要做研发，不论是客户端（当年是 Windows）还是服务器，包括一些分布式架构都相对经验多一些，希望是后续在自己不深入了解的领域可以多花些时间。

InfoQ：过去这一年自己最有成就感的事情是什么？感到自己在哪些方面又有成长了？

曾宪杰：去年一年，比较有成就感的事情有两个，一个是在我加入蘑菇街之后，在原先的技术团队的基础上，帮助团队的人员结构变得更合理了，不论在业务应用开发还是业务平台开发以及技术平台和算法方面，都引入了比较资深的 Leader 和核心员工，为整个集团业务快速发展提供了很好的技术人员储备，也能让技术跟上并超过业务的发展。另外一个事情是去年做了很多非功能层面的改造，使得当时的蘑菇街的技术体系有了比较好的分层和抽象，这也客观上让今年的融合在技术方面会更有准备些。

对于个人来说，过去一年自己负责的技术团队比之前的范围更广，过程中学到了很多之前自己不关注的东西，但是还需要再努力学习和提高。此外因为更多的参与整体公司战略、业务上的讨论，这些方

面也有很多收获。

InfoQ：关于蘑菇街的员工福利和执行力，我们听到过不少故事。您感觉这边技术团队的工作方式与工作氛围，有什么特别喜欢的地方可以分享？

曾宪杰：员工福利上，我觉得简单去比平均花在每个员工身上具体的钱，蘑菇街未必是最多的，比我们有钱的公司多了去了。但是如果说对员工的用心程度，我想蘑菇街是我接触过的公司里面最好的。举个例子，之前没有加入蘑菇街的时候，我也在自己业余时间受邀来跟蘑菇街的同学做过技术分享，临走的时候组织分享的同学给了我一个礼物，是一个蘑菇街自己的牙牙公仔，让我带给我儿子玩儿，因为来分享占用了我陪儿子的时间，后面回家把公仔给了儿子才发现袋子里面还有一盒香水是给我太太的，这些礼物从绝对价值上我想也许不算是多么多么的贵重，但我自己感受到的是组织分享的同学的用心；再比如说，蘑菇街是比较早就全员使用 Mac 笔记本的公司，在我看来这不是一个钱的问题，而是蘑菇街如何看待员工的问题，如何去给员工尽量提供更好的工作环境的问题。

而在技术团队上，毕竟自己之前也经历了很多，在团队中我强调的是技术团队之间是竞争和合作的，但是我们的合作是

大于竞争的，团队本身有自己的方向和分工，但是不论是团队还是个人，你都可以也有机会去挑战现有的实现方案，我们鼓励专业上的PK。大家做事情都比较简单。当然，也会遇到问题，但是我们直面具体问题，并且给明确的结论。不搞哲学讨论。

InfoQ：蘑菇街的技术博客 mogu.io 更新的挺不错，这个博客现在是怎样维护的？

曾宪杰：这个比较惭愧，之前带淘宝Java中间件的时候，我自己花了很多时间维护当时团队的博客。蘑菇街这里，我提到过我们应该去做公众号，应该有博客，但这个现在更多的是团队自己自发来维护。

InfoQ：蘑菇街的架构大量使用了开源软件，为了在业务中把它们用好用，就要把它们深入研究，并加入自己的修改。这边对于工程师把对开源软件的修改提交patch给社区是怎样的态度？是否会提供一些额外的时间和指导？

曾宪杰：对于开源，我的态度是合适的开源软件我们要首先拿来用，然后我们要研究他并且能用好，再接着是我们有能力来为开源贡献自己的力量，我非常鼓励工程师去提交Patch给社区的，15年Q4，我们数据平台的同学应该是提交了2015个Patch给社区。对于一些同学，去

做优化本身就是他的工作而不是额外时间才能去做的事情。

InfoQ：600多人的研发团队已经是挺不小的规模，而且蘑菇街现在还在做较多的招聘，尤其去年的校招好像规模特别大。现在校招和社招的新人进来会直接做一些重要的项目吗？接受怎样的指导？

曾宪杰：我们的项目很难说哪些重要哪些不重要，在我看来都很重要。校招和社招的新人进来后，会有师兄或者师姐来带，会逐步参与到线上的工作。这个过程中师兄/师姐和主管会给到员工很多的帮助、指导和要求，也会有研发部这里的新人通用的课程和专项的和岗位相关的课程。此外，我个人是觉得真正做事情是最好的成长和融入。

InfoQ：今年年初听说蘑菇街的架构在做前后端分离，以便更好的支持多个前端的同时保持后端的稳定。这是一个架构上比较大的调整，现在的进度如何？这个推进过程中有哪些关键的推力？有哪些大的困难？

曾宪杰：前后端分离的事情基本上我们改造结束了。其实比较大的困难是人员不够、业务不能明显受影响，然后还要把改造搞定。过程中在统一大家的认识、确定具体方案并且能确保方案合适蘑菇街的现状也花了很多时间。每家公司的人员情

况和当前系统的实现都不同，对蘑菇街来说，更主要的是希望前后端分离后，后端同学可以更关注业务逻辑，展现交互的逻辑就交给前端来做，中间通过纯数据的接口来交互，这可以很好的让双方都能够提升自己的效率。而推动这件事情能全部落地和搞定，在蘑菇街内需要我来做方向上的决策，然后拉相关同学来找到合适的方案，然后就是落地了。类似这样的事情，不能完全靠员工自发。

InfoQ：你们内部的 OpenStack + novadocker + Docker 私有云平台现在规模如何了？这方面现在主要在进行哪些工作？

曾宪杰：基于 OpenStack + novadocker + Docker 的私有云平台上线运行了一年多，这块已经相对稳定了下来。除了 Docker，我们线上也支持 KVM 的虚拟机，现在总体规模已经有了上千台。我们主要的目标是全虚拟化，也就是 Docker 和 KVM 能够覆盖整个集团的全部业务。现在我们的工作主要分三块，一块是保证线上的稳定性，及时解决平时出现的问题，保障每次大促的稳定性，第二块是不断完善现有的私有云平台，使得业务方能够更便捷的使用平台的各项功能，第三块是研发下一代基于 Docker 的容器云平台，目前主要在做的是基于 Kubernetes 的研发性工作，

同时也在升级 Docker 版本，跟踪业界最新的一些技术动态。

InfoQ：有些基础层面的技术，比如存储和 CDN，你们同时使用自己内部的资源和一些第三方服务。对于第三方服务的选择和采用，现在是以怎样的策略进行的？

曾宪杰：凡是能帮助蘑菇街业务更快、更好发展的第三服务，我们都表示欢迎合作，甚至主动寻求合作。合作伙伴的产品技术指标首先需要满足我们的业务要求与后期技术演进；其次我们也很在意合作伙伴能向我们提供快速响应的服务，甚至是主动服务；另外，选择和我们“趣味相投”，有同样的技术情怀、技术追求与技术积累的合作伙伴，也会有利于双方的顺畅合作。在同一领域内，我们通过技术、服务、商务等多纬度的对比与竞争，选择有效数量的合作伙伴为我们提供第三方产品与服务，这样一方面可以保障我们选择到更优的合作伙伴，另一方面我们也要让合作伙伴在与我们的长久合作中获取应有的回报，从而双方可以携手并肩、实现共赢。

InfoQ：技术团队在今年还有哪些重要的目标可以跟我们分享的？

曾宪杰：技术团队今年重要的事情就是平台化和混合云。去年年底，蘑菇街推

出了一个番茄炒蛋的 APP，这是当时我们除了蘑菇街这个 APP 外第二个面向消费者的 APP，当时我们的平台化做的还不是特别好，所以为了这个 APP 也做了一些重复的工作。今年希望可以在电商基础相关的用户、商品、交易、店铺、评价等等方面都平台化，可以更灵活快速的支撑上层业务，这个事情去年年底开始也在做今年会比较好的落地。混合云主要是来自于去年应对双 11 的思考。我们不能每年都通过

自己买很多机器的方式扛双 11 的峰值，所以必然是需要用公有云来帮助我们应对双 11 的一些峰值，当然不是在公有云上把我们的系统从头到尾搞一遍，更多的是分摊我们读的压力，真正的交易创建等等这些还是在蘑菇街自己的机房来做。而且除了双 11 的峰值外，我们也打算下半年的时候可以把更多的读利用公有云解决，并且是做到更小时间粒度的弹性。

关注微信号回复“**heron**”
观看符茂松在线讲解 Heron



专访符茂松： Twitter新开源的Heron是否可以完全取代Storm



作者 杜小芳

符茂松，Twitter 实时计算平台技术主管，负责 Heron, Presto 等服务。Heron 的原作者之一。专注于分布式系统，在 SIGMOD 等会议期刊发表多篇论文。本科毕业于华中科技大学，研究生毕业于 Carnegie Mellon University。

继 Storm 之后，Twitter 于今年 5 月份又开源了他们新开发的实时流式计算框架 Heron。Twitter 在 2014 年就用 Heron 完全替代 Storm，Heron 有很多架构方面的改进，而且向后兼容 Storm 生态系统。InfoQ 在第一时间联系了 Heron 的技术主管符茂松，了解一些技术内幕，看看 Twitter 的“鸟类文化”如何孵化了两代流计算平台。

InfoQ：现在开源 Heron 的原因？

符茂松：一是避免重复造轮子的代价。Twitter 的实时计算分析技术积累深厚；目前业界最流行的实时计算框架 Storm 就是我们开源的。我们针对 Twitter 在使用 Storm 遇到的痛点重新设计了新一代实时计算框架 Heron。很多重度依赖 Storm 的公司也有类似痛点并且希望可以使用 Heron。

二是吸引更多优秀的人共同建设 Heron。可以帮我们更好地理解需求并进行设计：事实上，近期在 Github 上也出现了很多我们忽略的需求。可以吸引到开源贡献者来做一些他们感兴趣，项目想要，但是我们本身没时间做的功能：目前除 Twitter 外，主要贡献者还有 Microsoft、Cisco、Adobe、Elodina、Stanford University、Indiana University……

InfoQ：如何判断软件开源时机成熟？

符茂松：

1. 项目本身应该是成熟的，稳定的，经过大规模生产检验的。Heron在2014年底已经完成取代 Storm成为Twitter新一代的实时计算框架；在日常生产中，输入流量的大小达到了10M events/s 的单个job运行起来毫无问题。
2. 项目本身应该有足够的文档。一方面，让第一次接触的人可以尝试运行；另一方面，需要提供项目的设计文档（已经各个自系统的文档），方便人们去理解去参与贡献项目。
3. 项目本身应该在一个状态。比如说，有足够的测试，足够的注

释，良好一致的编码规范，清晰一致的文件分布组织结构，等等。

4. 有良好的社区运作机制。比如说，有对应的GitHub Issues，Google Group方便交流；有相应的清晰的项目运作流程和规则，等等。

InfoQ：Heron 开发采用的什么语言，原因是什么？

符茂松：Heron 主要使用 Java 和 C++。

大部分的代码使用 Java 来实现的，因为：

1. Java代码的可移植性和平台独立性。编译后可单独运行。
2. 对于开发一个大型复杂系统，Java是一个相当简单易学的编程语言。Java项目也很容易管理。
3. 与其他语言相比，Java有巨大的开源库可用。
4. 比起C或C++，Java里灾难性崩溃更少。而且更易调试。
5. Java中的垃圾回收技术更容易复用或满足项目需求。

而 Heron 中的一些对性能和资源要求高的组件是用 C++ 完成的。

InfoQ：对于 Twitter 的两代实时处

理框架，Heron 最大的转变是什么？现在在流处理上是所有场景都可以用 Heron 替换 Storm ？

符茂松：在 Twitter 内部，我们的实时处理一直重度依赖 Storm。但是，随着数据规模增大，实时性能要求升高，以及案例的多样性以及数目的增多，Storm 的不足也就越来越明显。我们需要一个在共享集群设施中运行更稳定，更易调试，具有更好的性能并且易于管理的系统。考虑很多选择之后，我们得出的结论是需要构建一个全新的实时流处理系统。所以简单的说，Heron 和 Storm 最大的区别是性能可估，高效开发以及易于管理。

Heron 对于原生 Storm API 是完全兼容的，但是对于一些建立在原生 Storm API 之上的功能，基于用户需求和使用情况进行了裁剪。如果大家对这些功能有需求，我们也欢迎大家移植过来。

InfoQ：您对想研读 Heron 代码的人有什么样的指导或建议？

符茂松：我的建议是访问我们的官方网站“heronstreaming.io”，从“get started”开始阅读。

InfoQ：在设计流处理系统时，有哪些必须考虑的重要因素？

符茂松：理解需求，承认取舍，然后作出相应的设计。

对于流处理系统，以下需求（因素）的考量是最重要的。

首先是响应速度，作为一个实时计算框架延时一定要小。

其次是稳定性，可扩展性和性能可估性。实时数据流量一直是向上增长的；实时处理框架应该可以在数据增多的情况下工作稳定可靠，可扩展。

再就是易于调试。计算框架的出现，如 Hadoop 或 Heron，最重要的原因就是用来处理大规模的数据。

InfoQ：对于 Heron 或 Twitter 的流处理，以后有什么规划？对于实时流处理的技术趋势，您有什么样的看法？

符茂松：

1. 稳定、可扩展，功能更强的底层流处理框架。比如，我们在考虑加入以下功能的原生支持：如 stateful processing, precise recovery after failure (or so-called exactly once), runtime scaling。（更多详情请参照路线图）。
2. 使 Heron 具有高可拔插特性，并且我们为了这个目标已经做了很好的接口设计。我们希望 Heron 能满足所有需求，可以在任意环境下运行 Heron，如：Aurora, Yarn，

Mesos, AWS EC2等；也可以用任意语言来书写Heron job，如：Java、Python、C++等。

3. 在Heron之上此之上提供更加高级的接口，如SQL on top of Heron。这样的层次结构可以更好地协作，更好地设计，以及更好地优化。

InfoQ: Heron 的研发过程经历了哪些阶段？有哪些坑可以提醒其他朋友避开的？

符茂松：主要分成以下几个阶段。需要注意的是，以下阶段并不是有泾渭分明的间隔和计划的。他们是逻辑性的，实际上会混杂在一起，甚至交错。区分成不同的阶段更多是的事后反思得出来的结果。

1. 分析需求和思考设计目标

Heron 的出现是需求自然演化成产品的过程，而不是拍脑袋决定的。如之前所说的，我们重度依赖 Storm，但在使用过程中遇到了许多头疼的问题。从另外一个角度看，这些问题其实也是需求。我们意识到 Storm 已经不能够很好地满足这些需求了。我们整理出需求。需要注意的是，需求的出现并不是某一天某一刻我们一起开个会就产生的；而是基于运营维护的体验和教训，用户的反馈，自身的开发设计经验，逐渐产生的。

2. 调查研究

当 Storm 不能满足的需求越来越多的时候，我们开始思考解决方案。大体上，有三个方向：

- a. 继续改进 Storm
- b. 考察其他开源的实时计算框架，判断是否能够满足我们的需求
- c. 基于我们的需求，重新开发新一代实时计算框架

一般来说，这三个方向的难度是递增的；所以也会优先考虑前面的方向。我们持续改进 Storm，比如说使得单个 Storm Cluster 最大支持的机器数量有数倍的增长；同时我们也考察所有的开源实时计算框架。（2013 年）

我们发现：经过接近一年的优化改进 Storm 后，我们已经需要触及非常底层的设计。更深层次的优化改进，涉及的工作量剧增。

开源实时计算框架并不能够很好地支持我们的需求。考量主要是基于可扩展性，稳定性以及迁移成本。

这个时候我们开始认真思考重新开发新一代实时计算框架，并举行了多次会议进行讨论研究。Twitter 的实时计算分析技术积累深厚；目前业界最流行的实时计算框架 Storm 就是我们开源的。基于丰富的经验教训和确定的需求，我们确信开发

新一代框架是当时可以满足需求的最低成本的选择。

3. 设计、开发

与大部分项目开发过程类似，不赘述。从项目设计到第一个生产任务上线运行，大概使用了半年时间。（2014年4月～8月）

4. 迁移

Twitter 重度依赖 Storm，但是最大程度降低迁移成本也是我们首要需求以及设计目标。我们选择了兼容 Storm 原生 API，用户不需要更改任何一行代码；只需要更改编译文件，便可以把 Storm 迁移到 Heron 上。整个迁移时间预计 6 个月，实际耗时大概 4～5 个月。作为对比，我们当时对使用其他开源实时计算框架的预计迁移耗时是 16 个月。

经验教训：

1. 要理解需求，承认取舍，然后作出相应的设计目标。越早期的设计目标修改的成本就越大。很多时候，我们会想当然地进行设计，而不考虑是否真的满足了用户的需求。这样一方面不能解决实际问题；另外一方面会造成了资源的浪费。
2. 不要盲目重新造轮子。重造轮子的成本很大，应该是最后的选择。作为对比，我们组另外一个

服务 interactive query，在我们考察了市面上大部分的提供该服务的开源项目后，我们选择了 Presto 进行部署运维，并没有自己重造轮子，因为我们确信他能够很好地满足需求，是成本最低的选择；同时我们现在也积极参与到 Presto 的社区建设，把经验和代码回馈给 Presto 社区。

InfoQ：在 Twitter 工作是个什么样的体验？您能形容下你能感受到的硅谷文化吗？

符茂松：湾区公司的文化都是类似的，毕竟同一块土壤，同一群人，自然孕育出相似的结果。而 Twitter 的特点是“鸟类文化”：**敏捷，沟通，信任。**

Twitter 员工也许是湾区最了解鸟类的人了。Twitter 这个词本身就起源于鸟的叫声，而 Twitter 的 Logo 也是一只蓝色的小鸟。甚至，在 Twitter 内部文化都与鸟息息相关，比如 Twitter 欢迎新员工的标语就是“Welcome to the flock”（欢迎来到鸟群）；每间会议室都以鸟的名称命名；许多项目也以鸟的名称来命名，比如说，Heron。

敏捷：小鸟需要更高的振翅频率才能飞天。敏捷是在 Twitter 工作的态度和做事的方法：快速产出和持续迭代。相对于

直接给出完美的方案，我们更推崇先做一个可用的版本发给用户，看一下用户的反馈，基于用户的反馈再做下一次阶段的设计和迭代。

沟通：鸟聚成群。Twitter 的文化扁平，特别强调协作。人和人之间互相尊重，在项目开发过程中，鼓励在一起坐下来聊。对于重要的多组协作项目，还会临时开辟作战室，提供固定空间来让大家集中起来面对面讨论。团队协作讨论的时候，提倡平等的沟通氛围。日常交流也很平等，不会有任何等级差別的感觉。我刚入职作为新人的时候，就曾经给公司 CEO 写邮件提问问题，CEO 也会耐心地回复。公司日常会有非常多的课程以及培训，有大量的机会可以跟顶尖的人物交流分享。环境也是相当得开放，说句不好听得，像网吧一样。咋一看感觉会很粗糙，比如墙好像坑坑洼洼没有刷平，但是认真一看，会发现细节很是考究，比如说用的玻璃，桌椅，甚至白板空间结构物件摆放，都是经过精心设计的。开放式的环境也便于开放式的交流，有时候发现一个问题，可以轻松地招呼大家过来一起讨论。

信任：只鸟的力量有限，鸟儿互相依赖信任对方。信任，开放，是我进入 Twitter 第一天的最大的感受：可以看到大量实时的统计分析数据，比如说不同产

品的在不同地区的实时活跃用户，不同产品灰度测试结果；代码库是所有工程师都有权限完全查看的，如果你对某个项目特别有兴趣，完全可以直接把代码下载到本地阅读学习。同样，Twitter 也欢迎员工平时带朋友和家庭成员来公司玩。

专访链家网前端总架构师杨永林： 8年架构师成长之路



作者 韩婷

杨永林，人称“教主”，8年前端开发经验，原新浪微博前端技术专家，现任链家网前端总架构师。长期研究 Web 访问性能优化和前端框架搭建。

作为初始团队成员，教主参与了新浪微博所有 PC 版本的开发，其中 4 ~ 6 版以架构师的身份设计了微博 PC 版的前端架构。在新浪微博任职期间，教主设计实现了流水线加载技术与模块化代码组织，达到了在提高访问性能的同时极大降低了开发成本的目的。主要研究方向是 Web 访问性能优化与框架组织。在国内为数不多地实现了 BigPipe 技术，极大地提升了微

博的访问速度。同时，微博的前端代码基础包、前端框架和构建工具均出自教主之手。

2015 年年底，教主加入链家网，负责前端的整体架构工作。

8 年的前端开发生涯中，教主是如何一步一步地成为知名前端架构师的呢？为何选择加入了链家网呢？带着这些疑问，InfoQ 采访了这位智慧而幽默的架构师。

InfoQ: 您在微博和链家都是前端架构师，能说说前端架构师这个工种具体是做什么的吗？

杨永林: 我对架构师所担任的职责的认识是一步步变化，慢慢深入的。

在刚参加工作的时候，我觉得架构师就是代码写得又快又好的人，是工程师的晋级版本。

工作过一些年以后，我发现仅仅提高自身的开发效率是远远不够的，团队需要整体的提升。发现这一点后，我开始制作并完善各种开发工具，编写开发框架。

最近几年，随着迭代开发了一些产品本，我又发现之前能够提升效率的框架工具很有可能在后来成了产品发展的绊脚石。这时，我开始考虑架构设计的指导原则，开始考虑取舍。一些在短期内能够提升效率但不符合原则的东西，我就选择不做或者想办法在原则的指导下进行改进。比如我相信可变化的代码才是有生命力的代码，在架构设计上我也会趋向于让项目的代码可以一点一点的变化演进，不是那种一言不合就重构到状态。所以我认为前端架构师就是那种在前端领域提出开发的指导原则，在原则下设计开发框架和开发工具，让更多的开发者可以协同工作的人。

InfoQ: 您在新浪微博的时候设计了前端架构，能否介绍一下包括了哪些组成

部分，有什么关键技术？

杨永林: 主要是代码基础包，页面加载框架和前端构建工具。

早期前端开发面临两个主要问题是浏览器兼容和 API 不够丰富，基础包一般都是用来解决这两个问题。当时新浪有一个自己的 Sina 包，但是代码比较零散，模式也不统一，各产品线有自己的扩展，同样的功能可能有多种实现，不太好维护。后来我用业余时间开发了一个带有命名空间管理功能的基础包，特点就是简单清晰，易于使用，被团队采纳作为了微博的基础包使用至今。

页面加载框架是被需倒逼着产生的，2010 年微博业务膨胀，页面展示的内容越来越多，这使得页面响应速度也变得越来越慢。我所在的团队接到的需求是要求在内容变多的情况下将响应速度变得更快。

这个时候 Facebook 推出了 BigPipe 技术，我们觉得这个理念正好能够解决我们应对的问题，所以决定实施，但当时 Facebook 只是分享了他们的做法，并没有提供实现，所以对我来说也是巨大的挑战。我当时将页面划分成多个独立的子模块，模块是完全可以自主运行的，模块可以嵌套，所以页面就是一批模块的树形堆叠。服务端用 Chunked 的方式将模块的信息以 JavaScript 代码块的方式传输到页面，而

前端需要做的很重要的工作是管理每个模块的生命周期。

我很荣幸那时能有机会和团队成员一起开发了这个加载框架，我们可能是国内第一个在大型互联网应用上全面使用这项技术的。之后的一年我一直致力于此项技术的优化工作，比如支持服务端乱序输出，保证服务端可以使用并行策略，压缩，减少前置依赖条件等，并在 2013 年与 @Laruencc (鸟哥) 合作实施了 CBigPipe (并行的 BigPipe) 技术，进一步提高了这项技术的性能。微博的 V5 版的加载性能也达到了顶峰，页面的加载速度几乎相当于静态网页。

前端构建工具是这几年才开始流行，其实早在 2008 年的时候，新浪就已经使用前端小文件开发，使用构建工具进行开发，测试和上线。现在想想应该还是比较超前了，不过那时的版本是需要 PHP、Python 和 Java 环境，团队维护起来比较困难，而且使用的是字符串替换方案，功能比较有限。2012 年我将这个工具进行了改造，使其仅需要 Node 环境，同时支持开发、测试部署和打包上线。由于使用了 UglifyJS，有了语法树，我加了一些以前没有的功能，比如预编译的模版引擎、支持模版嵌套和母模版、代码健康度检测、冗余模块分析等。

InfoQ：前端构建工具前后有 Grunt/Gulp、Webpack、npm scripts 等，您对这些工具有什么看法，哪个更好？如何选择适合公司产品的工具？应从哪些方面考虑？

杨永林：我觉得这些工具有效地解决了前端开发效率的问题，它们的出现都是对技术的推动，如果在我做工具的时候有这些项目的出现，会减少我很多的工作量。至于哪个更好，我觉得，你能掌握哪个，哪个就是最好的。因为说到底，工具是为你的业务服务的，你可能需要对它做些改造或者是写一些扩展，在这个时候你发现你对他的熟悉变的很重要。构建工具的迁移成本还是挺高的，我不太推荐频繁地变更它，所以最好不要追着流行走，还是要根据自己团队的特点，因地制宜地选择一款合适的。如果不是超大型的应用，其实构建的结果的影响并没有太大的差异，与其想着哪个更好哪个更牛逼，不如将其中一个玩熟玩透。

InfoQ：如何保证团队成员不会踩到同样的坑？在设计框架和构建工具时有无这方面的考虑？请举例说明。

杨永林：首先，制定规范、分享经验是免不了的，但纸上得来终觉浅吧，很多时候，亲身踩一次坑，得到的经验才会深刻。而我所要做的是在团队成员踩到坑的

时候降低这件事造成的后果。比如我提供的开发环境是可以完全模拟线上环境的，测试代码和线上保持一致，很多意外情况都可以在开发、测试期被发现。同时，制定的开发规范要由工具检测来保证，不符合规范的代码不能够打包上线。对于规范代码可以使用工具计算出业务影响范围，能有效保证测试覆盖面。总的来说，踩坑不要紧，架构来帮你兜底，爬出坑的过程就成为了团队成员所得到的财富。

InfoQ：您认为对 Web 访问性能的优化需要关注哪些方面？其中，最值得关注的点是什么？为什么？

杨永林：我觉得性能优化需要方方面面都要兼顾，包括网络时间、服务器计算时间、页面请求数、下载量、页面载入模型等。而这里面任何一项的性能提升可能都需要你修改大量代码或者调整架构来实现，但是得到的效果可能就是一点点。因此很少见到银弹，一般都是一点一点地做出来的。我这里谈两个我觉得比较值得关注但很容易被忽视的点吧。

一是你所服务产品的形态，用户关心什么，这是一些工程师比较容易忽略的。有些产品需要用户打开时很快，有些需要用户使用时流畅；有些产品用户可以容忍看旧数据，而有些则必须是新内容；有些产品用户一天打开很多次，而有些看一次

就关掉了。这些产品需求的差异都会影响你的决策。

二是评测标准，用什么来测量性能的好坏。一些人认为请求数或者请求量减少了，访问就快了，其实这是不一定的，有可能你花了很大精力做的事情在用户看来并没有什么太大变化。所以，找一个评测标准让每一个优化在数据上有所体现是很重要的。

InfoQ：度量前端性能的指标有哪些？如何对 Web 访问性能进行监控？

杨永林：我所服务的产品一般都关注访问性能，也就是用户看到内容的快慢，所以我们一般用首屏时间来评估，一般的性能检测服务商都能提供这个指标。

选这个指标有两点考虑：一是因为它并不是一个技术指标，而是一个感知指标，所以更接近人类的感受。二是旁路检测，它并不在系统内，不是系统汇报上来的数据，这样就有效的规避了幸存者偏差的问题。当然它也有些不足：一是数据采样小，二是可以被欺骗。所以可能需要一点儿统计学功底和性能监控的正确认识。

在监控的过程中，一是要关注长期趋势的变化，如果不是突发状况，单点的数据的绝对值是没有意义的，要收集长期的数据，分析其中的变化，当有变更的时候尤其要关注数据的变化。二是关注最差

25% 的状况，有些人，会在公司内网刷自己的产品，感觉挺快，其实不论你用什么手段，只要网快，用户的体验都不会太差，体验的差异在于最差那部分用户。三是从不同维度分析数据，如地区、网络、时段、运行环境等。

InfoQ：前端工程师如何成为前端架构师，除了编程能力和架构知识，还需要培养哪些能力？

杨永林：我想，大部分领域的架构师工作都是差不多的，就是搭建一个解决问题的框架，让团队成员能在框架下良好的配合工作，完成产品的开发需求。

我们知道，解决一个问题的手段有很多，在这个过程中取舍就很重要了，我们也知道，没有银弹，很少能遇见那种全面优势的解决方案，大部分方案都是牺牲掉一部分东西来换取一部分东西。因此，作为架构师，不仅要对各个技术方案的特点、成本要熟知（也就是编程能力和架构知识），还要学会如何选择。显然，架构师需要根据产品的特点和发展方向做出决定，在前端领域的架构要能让配合的团队对接的顺畅。那么在这个过程中，良好的沟通能力、同理心、利他的思维方式，就显得很重要了。因为我们不仅要完成开发任务，也要思考在自己的领域内如何帮助项目解决问题。

InfoQ：据说有些同事在对技术的讨论中以“击败”您为荣，您是如何看待的？这对团队及其个人的发展带来了哪些影响？

杨永林：这是我一个毛病，喜欢给别人方案着茬。我觉得这是一个思辨的过程，通过从不同角度分析问题，去挑战解决方案的合理性，才能让问题解决的更稳妥。在知识的获取中也是这样，一次又一次地去问为什么，去追根溯源，才能让知识体系更牢固。

我很喜欢在团队内扮演一种“反派”的角色，从反面的角度分析问题，去挑战别人的方案。其实，我不是真的去否定他，而是希望他的方案是经过反复推敲、深思熟虑产生的，这样的方案会更健壮。时间长了，他们会觉得我是一个爱抬杠的人，就会做足准备来“挑战”我。能把我说得接不上话来，他们会觉得很开心。这个结果是我想看到的，因为这说明团队成员在解决问题时进行了充分的思考。

InfoQ：您为什么放弃了在之前新浪微博的元老级身份，而选择加入链家网？

杨永林：这可能源自我对工作的看法吧，我觉得人生活在这个世界上，工作是在为社会创造价值和财富，这和他具体从事哪种职业没有直接关系。现在行业里有一种风气，就是觉得程序员写好代码就好了，

不用关心自己做的事情是什么。甚至社会上也给程序员打一些什么“木讷”、“情商低”之类的标签。我觉得不应该是这样的，程序员也是社会人，也有他的社会责任，也有家庭责任，也需要陪伴他的伴侣，照顾他的小孩，不是每天只是面对代码而不管其他的事。人不要因为群体印象就把自己限制住，人的生活就应该是多种多样、丰富多采的，人生应该是有意义的。

就我个人而言，在过去的几年，我所服务的产品不仅加深了人们之间的沟通和理解，也使得国家的信息变得更透明。而我所做的工作对这样的产品做出了贡献，可以说我的工作让世界变得美好那么一点点。这让我觉得我的人生增添了那么一点意义。而当我搭建起前端框架后，我个人能起的作用变得越来越小，我能继

续创造的价值也越来越少，所以需要另一个平台来继续发挥我的能量。

这时我有机会接触到链家网，这家公司致力于解决人们的居住问题，它让中国最大的市场变得透明、有序。我觉得链家网做的是很有意义的事，同时，它仅仅用了不到两年的时间，就集结了一批各领域的牛人，维护了国内规模最大的房地产交易系统，用技术手段让房屋的买卖变的更轻松、透明、快捷。在与链家网的接触中，我感受到了那种积极解决问题的活力和务实做事的态度。再加上链家网中大部分技术人，在之前也都是各个大型互联网公司的中坚力量，我想没有什么比与志同道合的人来一起改变世界更令人激动的了。此时，鸟哥专门来邀请我加入链家网，我就毫不犹豫地同意了。

关注微信号回复“桐木”

看阿里 10 年开发专家预测前端未来





▶ About StuQ /关于StuQ

StuQ 是极客邦科技旗下 IT 职业教育业务，通过实践驱动的系统课程，帮助 IT 从业人员提高职业技能，获得更好的工作机会。

▶ Online Course /在线课程

StuQ 不仅邀请业内顶尖的技术专家录制视频课程，还会为用户在学习环节提供项目实战演练，技能测试，工作机会推荐 3 个课程服务。

▶ Micro Class /微课堂

讲应用，重实战，强互动，打开微信收获一堂干货课。

有干货，有兴趣，随手拯救技术宅，轻松报名微课堂，免费获取技术大会 PPT。



StuQ 微信公众号: StuQ

▶ OUR CHARACTERISTICS /我们的特色

一线专家



远离教书先生,我们邀请行业里的技术专家,将当下最需要技术讲解给大家。

技能评估



课程中, 讲师将会以实际项目, 对学员的学习成果进行评测与考核。

项目辅导



通过参加我们的课程项目, 在实践中进行学习, 提升解决项目中具体问题的能力。

专家认证



通过参加StuQ专家顾问团的面试, 获得来自专家的技能认证推荐信。

一线专家团队驱动的企业培训服务



Enterprise Training

同国内外企业深度合作，为其设置专属定制化学习课程，紧密贴合用户个性化需求的内容与形式，提供top级的学习体验。

Deep Workshop

携手全球知名一线技术专家团队分享软件研发技术管理实践，提供研发团队提升必修精选课程。

Open Class

- 工作坊学习体验、案例干货尽收
- 大时段分享
- 全程30个精选课程研修
- 纯净绿色学习环境
- 情景教学、沙盘演练
- 以本土化选题适应城市风向需求

博文视点

@博文视点Broadview

博文视点Broadview

IT出版旗舰品牌



一本围绕实战讲解Kubernetes
如何落地的书



一幅完整的App后台开发蓝图



移动开发者学习React Native
不可多得的入门宝典



腾讯互娱倾情力献,
腾讯游戏系统揭秘



物联网时代颠覆安全最前线



还原比特币外广衰真相



给予你创造引人入胜的游戏
所需要的一切技巧和工具



互联网+垂直领域创业必读书



首度聚焦安全风眼

EXPERTS RECOMMEND

专家推荐

▶ 程立 / CHENG LI

持续关注InfoQ好多年了。由于工作繁忙没有很多时间泡技术社区，我一直选择坚持精品与原创路线的InfoQ作为获得业界信息的主要来源。当遇到难题时也会到InfoQ上寻找灵感并常常有所收获，可以说InfoQ是我的老师、智囊和朋友，借此机会向InfoQ说声谢谢！

▶ 冯大辉 / FENG DAHUI

InfoQ，技术人都喜欢。几年下来，通过InfoQ网站获得了许多有价值的资讯，通过InfoQ的电子杂志借鉴到很多技术思路，而通过InfoQ举办的数次QCon大会，又结识了不少业界朋友。期待InfoQ坚持自己的特色，期待越办越好！

▶ 洪强宁 / HONG QIANGNING

InfoQ是我获取业内最先进的技术和理念的重要渠道。在InfoQ的帮助下，我也得以与国内外众多技术高手交流切磋，获益匪浅。感谢InfoQ！

▶ 卢旭东 / LU XUDONG

我很早就是InfoQ的注册用户了（哈哈，有好几年了吧，持续保持潜水状态），它一直是了解业界研发趋势，学习先进技术和方法的最好平台！在这里还能认识很多志同道合的朋友，InfoQ有潜质成为国内最专业、最大、最有影响力的研发社区！InfoQ的电子杂志更是必看，深浅结合，对实践很有指导性。

▶ 王文彬 / WANG WENBIN

InfoQ办的QCon大会是一个高质量的盛宴，对于最新的互联网技术和最佳实践一直在做探讨。除了邀请国内的牛人，也会有国外的大牛来做分享，对技术人员是一个不可错过的大会。

▶ 杨卫华 / YANG WEIHUA

InfoQ每年遍布全球的QCon大会是技术界的盛会，给业界很多研发方向上的启发，新浪微博的技术架构也从往届QCon大会演讲中获取了不少宝贵经验。

▶ 吴永强 / WU YONGQIANG

接触InfoQ，包括QCon，已经有好几年了，我非常喜欢它的风格，灵动、快速、实用，Moq网站、QCon、《架构师》杂志都能够紧贴互联网技术的发展前沿，带来大量的最佳实践，对我们这样发展中的公司的帮助非常大。希望InfoQ能够越做越好！

▶ 毛新生 / MAO XINSHENG

InfoQ社区是架构师的一流资讯来源，也是大家交流的桥梁。