加长版

建一支强大的小团队

陈皓

2012年10月

个人简介

- 行业背景
  - 金融行业(Thomson Reuters)
  - 计算平台(Platform)
  - 电子商务(Amazon)
- 技术背景
  - C/C++/Java
  - Unix/Linux/Windows
  - Web

Weibo: @左耳朵耗子
Twitter: @haoel
Blog: http://coolshell.cn/

我的个性

- 码农兼包工头
- 敏捷恐怖分子
- Unix/Linux/C/C++脑残粉
- C2C痛恨者
- CSDN腾讯百度批评人
- “技术部门无技术种族”歧视者
- “程序员文化”民族主义者

前言

- 国内的IT文化和国外的IT文化在认识上差距很大，就像我早几年宣扬Unix的文化的时候，国内还沉浸在微软的文化中。我们在关注一些技术和公司的时候，都忘了关注相关的文化，这是我想分享的原因。

- 理想是丰满的，现实是骨感的。在中国，这些东西可能还比较理想，我们现在客观上还要向现实妥协，但我希望以后走上领导岗位或要开创自己的公司的朋友能坚持住这些理想，不要向现实妥协。

- 下面的文字有粗口，请确定你已成年。

软件开发现状

软件开发的现状

- 软件的复杂度持续不断地提升
  - 业务需求复杂度
  - 部署运营规模复杂度
  - 维护支持复杂度
- 软件开发迭代周期和频率越来越快
  - 开发、测试周期
  - 交付周期
  - 解决问题的周期
- 软件的运行和质量要求的越来越高
  - 扩展性
  - 稳定性、可用性
  - 用户体验

软件开发出现的问题

- 软件的问题
  - 业务分析能力的不断提升
  - 软件产品的质量随着复杂度的提升越来越差
  - 软件的测试和维护成本越来越大
  - 解决软件问题的成本越来越大

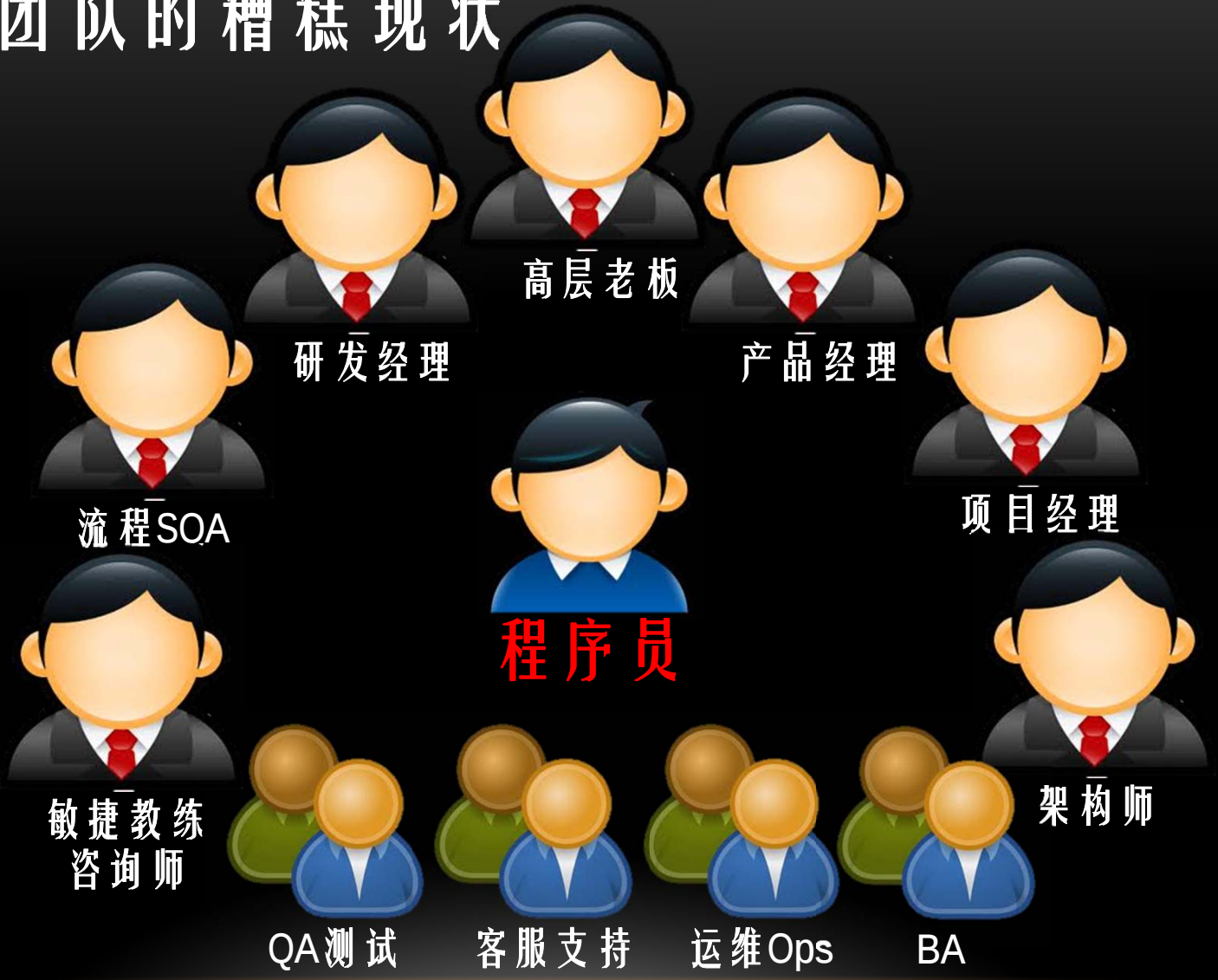
- 管理问题
  - 团队越来越大，分工越来越细
  - 团队的合作问题
  - 团队的执行力问题
  - 团队的流动问题
  - 团队的成长问题

如何解决?

软件开发出现的问题

- 软件的问题
  - 业务分析能力的不断提升（加入BA和PM）
  - 软件产品的质量随着复杂度的提升越来越差（加入咨询师、QA和SQA）
  - 软件的测试和维护成本越来越大（加入QA、Ops和流程）
  - 解决软件问题的成本越来越大（加入咨询师、各种图表式流程）
- 管理问题
  - 团队越来越大，分工越来越细（人海战术，头痛医头，脚痛医脚）
  - 团队的合作问题（利益牵扯，互相扯皮推诿）
  - 团队的执行力问题（流程太重，利益牵扯，互相扯皮，人浮于事……）
  - 团队的流动问题（流程太重，重复劳动太多，光说不练的越来越多，加班重，内耗大，政治多……）
  - 团队的成长问题（沟通复杂，内耗越大，重复很难成长）

软件团队的糟糕现状



每一个苦逼的程序员背后，都有一群指点江山的人


销售经理
市场经理
人事经理
安全经理
保安队长
公关经理
项目经理
QA经理
技术经理
产品经理
me !

问题

- 流程控制多
- 组织单元多
- 利益牵扯多
- 消耗资源多
- 沟通会议多

这样的IT公司和天朝的"XXX局"没有什么差别

思考

- 程序员不能管理项目和进度?
- 程序员不能做测试保证软件质量?
- 程序员不能分析需求?
- 程序员不能做运维?
- 程序员不能管理好自己的流程?
- 程序员不能架构和设计软件系统?
- 程序员不能和别人很好的合作?

再思考

- 如果不信任程序员，你还招他干啥?
- 如果程序员只会编码，这是不合格的程序员!
- 自己管理自己是最有效的!
  - 管不好自己和自己工作的员工，找人来管是错误的!
- **程序员是生产力的主力!**
  - 其它大多数人都在降代团队的生产力
  - 只有程序员在commit软件中那一行一行的代码

程序员的必备技能

- 程序员必需懂软件开发
  - 编程技术，软件设计，开发工具，软件工程（流程）
- 程序员必需理解需求
  - 必需站在用户的角度上思考问题
- 程序员必需懂测试
  - 自动化测试，测试案例设计
- 程序员必需做系统维护
  - 自动化部署，自动化监控
- 程序员必需管理好自己的工作
  - 程序员知道什么流程方法最适合自己
  - 知道自己做的轻重缓急和时间进度

程序员是三头六臂?

- 当然不是，因为这些都是必修课
  - 懂分析软件需求
  - 懂软件过程
  - 懂设计软件
  - 懂编码
  - 懂测试软件
  - 懂维护软件
  - 懂得管理自己要做的事

- 不懂这些，怎么可能写好程序?

一些重要观点

- 优秀的程序员创造的价值是平庸程序员的10000倍！
  - Unix/Linux作者创造的价值 = 在App Store写某个App价值的N倍（N > 10000倍）
- Eat Your Own Dog Food! 只有吃自己的狗食才会有改进！
  - 吃狗食还不够，还要吃自己的Dog Shit!
  - 自己写的软件自己测试，自己运维，才能亲身体会难点在哪里
  - 只有亲身体会痛苦，才会有想要改进的动力
- 程序就是用来自动化一切机械的劳动！
  - 只要你发现你在干重复的事情，你就开始要思考用程序来帮助你
  - 只要你发现你在干劳动密集型的事，就要开始担心被机器所取代
- 主动工作的生产力远大于被动工作的生产力！
  - 自己想干的事 远比 被逼要干的事要干得好！

GOOGLE的工程师文化

- **对工程师的信任和尊重**
  - 每位员工都有20%的工作时间可以做自己想做的事情，很多产品就是用这20%的时间做出来的，比如众所周知的 Gmail。
  - Larry Page:“this company will be run by engineers” (not marketing not sales) 这句话是Larry Page在Google进行IPO前对董事会说的
- **自由平等，信息透明**
  - 数据说话 (*If we have data, let's look at data. If all we have are opinions, let's go with mine* - Jim Barksdale, former Netscape CEO)
  - 层级扁平，TGIF上可以提各种尖锐问题 (dory and live questions)
  - 公开所有项目的：文档，代码、规划，数据，资源
- **自己设置目标**
  - 激进地目标设定，完成基本上是60%左右。
- **给管理层打分**
  - 不受程序员喜欢的管理人员会被离职。

注：Eric曾经用PM掌权，结果基本就是模仿，如Android, 和 Google+。Larry回来后，工程师重新掌权。

FACEBOOK的工程师文化

- 项目的资源完全来自工程师的自愿
- PM游说工程师，试图吸引工程师为他们的想法而工作
- 工程师自己决定去干哪个产品经理的活，工程师的头儿几乎可以说是放任手下各行其是
- 工程师自己处理所有的事情。从js到db，并可以得到一些UI/UX设计师的支持。通常说来，工程师干所有的活。
- 一个工程师能够serve一百万的用户

引自：How Facebook Ships Codes
http://framethink.wordpress.com/2011/01/17/how-facebook-ships-code/

AMAZON的工程师文化

- SDE 被戏称为 Someone Do Everything
- 工程师从需求分析一直做到线上运维（还有招聘）
- 大量的自动化工具（测试，部署，监控）
  - 这就是程序员吃狗食吃出来的结果
- 需求来的时候要问为什么要做？意义何在？
- 少量的QA，Designer，没有架构师
- 没有特别的软件开发流程，团队决定用什么流程
  - 很多团队使用Scrum的部分实践
- 非常严格的招人
  - 和Google一样，新人要超过现有一半以上的人
- Pizza Team 文化
  - 2 Pizza Teams，两张Pizza可以喂饱的团队。
- Up the River
  - 公司内顶尖工程师讨论出的项目可直接Assign给SVP。

注：因为Amazon不像Google和FB，它有有大量的线下业务，所以，还不能算真正意义的工程师文化，很多团队还是业务驱动型的。是各种文化的杂交体。

工程师文化 == 创新冲动

非工程师文化会是什么结果?

- **流程文化** == 官僚主义+办公室政治
  - 劳动密集型企业，如：富士康
- **咨询师文化** == 假大空文化
  - 大量的概念和各种前景
- **产品经理文化** == 模仿抄袭文化
  - 谷歌的Eric Smith的PM文化干出的Android和Google+都是模仿，Larry Page上台后干掉了PM的SVP，隔离了所有的PM，让他们report给了Engineering 团队
- **老板文化** == 独裁文化（要么牛，要么混蛋）
  - 老板牛，什么都牛，老板混蛋，全面混蛋
- **营销文化** == 腐败+弄虚作假文化
  - 各种急攻近利的营销手段

软件公司的两种管理方式

摘自StackExchange

详见：http://coolshell.cn/articles/4951.html

WIDGET FACTORY（小商品工厂）

- 一般人的本性是懒惰的，工作越少越好，可能的话会逃避工作，大部分人对集体（公司，机构，单位或组织等）的目标不关心，因此管理者需要以强迫，威胁处罚，指导，金钱利益等诱因激发人们的工作源动力。
- 公司对实实在在的软件开发漠不关心，他们想要把理想中的软件开发运作变成他们看得见的图表。
- 流程很重，组织单元和人员很多。

FILM CREWS（电影工作组）

- 这种公司认为人是有相当高的智力和创造力的，是自己可以激发自己的。
- 每一个个体的自己专业能力，要远远优于那种被组织和协调出来的能力。人们努力工作，并且可以享受工作。
- 工作职责变得非常地垂直——你需要具有从上到下的而比较宽泛的各种能力。
- 经理了解把一个伟大的软件组合起来的每一个碎片。他的角色是鼓舞大家，守护着方向（Vision）
- 团队里的每一个人都很关键，因为团队相信软件的结果来自所有的参与者，以及他们的那种独一无二团队工作方式。

小团队的前提——工程师文化

- 创始人是工程师（什么样的人什么样的团队）
- 知识（技术）密集型的企业（有强以抑制的创新冲动）
- 价值观和目标一致（这点非常重要，所谓culture fit）
- 资源平等，信息透明（只有理解一致，互相补位，才可能是小团队）
  - 总线宽度宽，群体智慧，共同负责，鼓励越界工作
- 雇佣最好的人，个体能力强，每个人都是Leader（对从事的事情不但有热情）
  - 不只是相互领导，而是相互协作
  - 没有中心控制，通过团队协作自治
  - 自组织，自协作，自管理，自进化
- 简化功能，抽象软件（如：big table, v8, HipHop, Hadoop, ...）
- 残酷无情地推动自动化（剔除重复劳动）
- 残酷无情地降低维护工作量（剔除无创新的工作）
- 允许20%的自由时间
- 维护一个相互尊重，不断反思和相互学习的环境
  - 只有痛苦才会让人反思，只有大家都强，才会互相学习

没有银弹

怎么培养工程师文化?

- 什么样的领导，什么样的文化
- 什么样的创始人，什么样的公司
- 推荐阅读
  - Quora：What makes a good engineering culture?

    http://www.quora.com/Software-Engineering/What-makes-a-good-engineering-culture
  - Google 的 Wayne Rosing 访谈

    http://www.youtube.com/watch?v=pLj3qSXNy7o （part 1 of 9）

谢谢！