建一支强大的小团队

陈皓

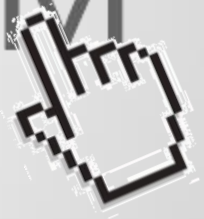
2012年10月

更多资讯，点击访问：



腾讯大讲堂

DJT.QQ.COM



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

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
  - 业委需求复杂度
  - 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

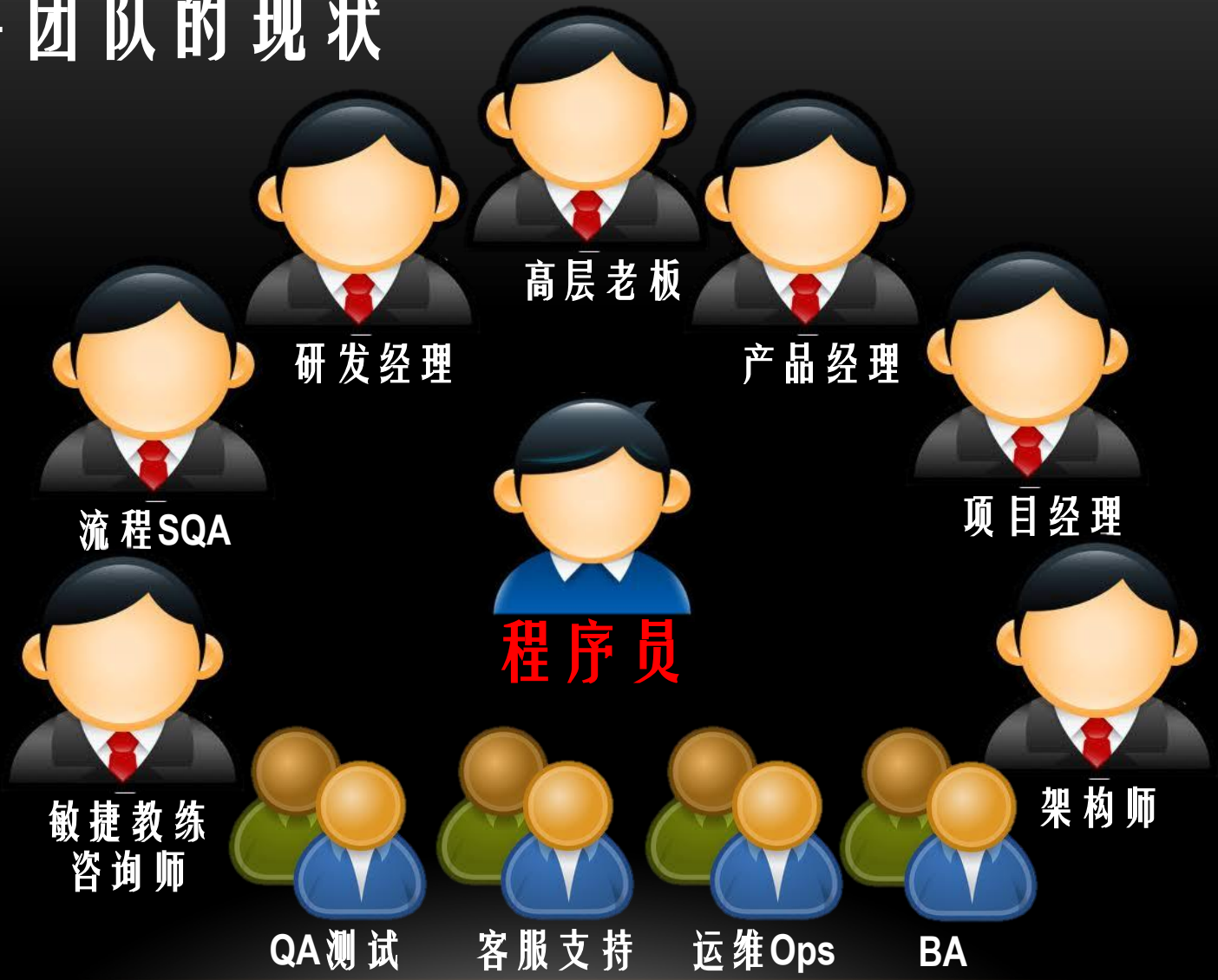
- 软件的问题
  - 业务分析能力的不断提升
  - 软件产品的质量随着复杂度的提升越来越差
  - 软件的测试和维护成本越来越大
  - 解决软件问题的成本越来越大

- 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（内部和外部）
  - 团队的成长问题



如何解决?

软件团队的现状



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 !



Talk is Cheap, Show me the Code !

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



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?

- 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

- 优秀的程序员创造的价值是平庸程序员的10000倍！
- Eat Your Owe Dog Food! 只有吃自己的狗食才会有改进！
- 程序就是用来自动化一切机械的劳动！
- 主动工作的生产力远大于被动工作的生产力！

程序员的必备技能

- 程序员必需懂编程技术，开发工具，软件工程
- 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
- 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
  - 轻重缓急，时间进度



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)
- **自由平等，信息透明**
  - 数据说话 (quote netscape executive)
  - 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。

FACEBOOK的工程师文化



- 项目的资源完全来自工程师的自愿
- PM游说工程师们，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
- 工程师然后去给他们的头儿报告：“我本周要干5件事情”
  工程师的头儿几乎可以说是放任手下各行其是，偶尔给点做事情优先级的忠告。
- 工程师自己处理所有的事情，从js到db的所有逻辑。如果他们期望得到设计师(FB里只有非常少的专职设计师)的帮助，他们需要自己去搞定设计师来加入他们的项目；如果需要架构师同样也得自己来搞定。但通常来说，工程师自己干所有的活

AMAZON的工程师文化



- SDE – Someone Do Everything
- 工程师从需求分析一直做到线上运维
- 大量的自动化工具（测试，部署，监控）
- 需求来的时候要问为什么要做?
- 少量的QA，Designer，没有架构师
- 没有软件开发流程，团队说得算
- 非常严格的招人
- Pizza Team 文化
- UP the River



工程师文化 == 创新冲动

软件公司的两种管理方式

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。
- 经理总是要去做他下属的工作。这种团队中的每一个人都很容易被别人取代，在这种团队里，经理的工作能力不断地被加强
- 这种公司一般使用树形层级的组织结构，而不是水平式的工作角色。
- 他们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。

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

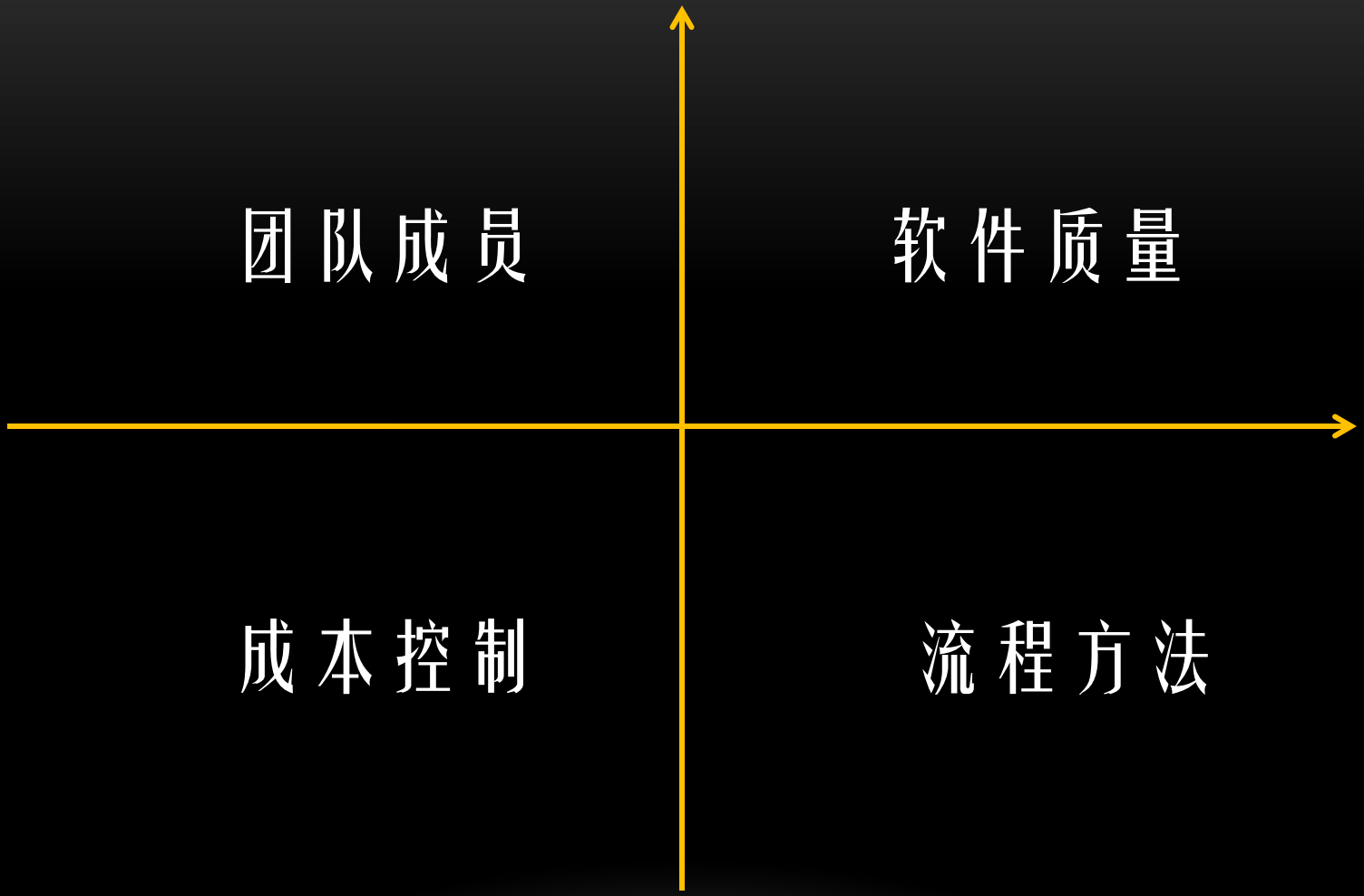
- 知识密集型的企业
- 价值观和目标一致
- 产品要简化到简单到不能简单
- 资源平等，信息透明
- 个体能力要强，都是T型人才
- 没有中心控制，通过团队协作自治
- 自组织，自协作，自管理，自进化



软件团队管理

团队管理三板斧

- 标准化流程（制度）
  - 质量控制
  - 过程管理
  - 项目监控
- 过程改进（方法）
  - 持续集成和自动化测试
  - 极限编程
  - 快速迭代
- 激发团队和人员（文化）
  - 团队和个人发展
  - 团队主人翁精神
  - 团队和个人的执行力


团队成员
软件质量
成本控制
流程方法

谢 谢！