博士研磨

一个博士研究生的回忆录 [1]

原著　Philip J. Guo　(philip@pgbovine.net)
翻译　齐鹏　(pengqi@cs.stanford.edu)

[1]英文原文链接:http://www.pgbovine.net/PhD-memoir/pguo-PhD-grind.pdf。本翻译基于原文2012年7月16日发布的版本译成。

献给所有热爱创造的人。

目录

Disclaimer

This is an unauthorized Chinese translation of Philip J. Guo's memoir The Ph.D. Grind, and the original author did not have any input on the translation.

Apart from the Translator's Preface, none of the content of this work represents, none should be interpreted as, the opinion of the translator; nor will the translator be responsible for the consequences of any interpretation of this translation.

声明

本作品是Philip J. Guo的回忆录《The Ph.D. Grind》的中文翻译，翻译并没有得到原作者的任何授权，原作者也并没有以任何形式参与到翻译过程中。

除译序外，本作品的一切内容均不代表——且不应被认为是——译者的观点；译者对此翻译的任何理解以及其导致的影响亦不负有任何责任。

序

这本书记述了从2006年到2012年，我在斯坦福大学攻读博士研究生期间六年的求学经历。这本书适合广泛的读者群，其中包括：

- 有志攻读博士研究生[1]的本科生；
- 寻求方向或灵感的在读博士生；
- 希望更深入了解博士研究生的教授；
- 希望聘用和管理拥有博士学位员工的雇主；
- 在充满竞争的创新领域工作、与自我追求和自我激励密不可分的专业人士；
- 对学术研究充满好奇的有一定教育背景的成年人（或者早熟的青少年）。

《博士研磨》与已有的与博士经历相关的文章在写作形式、写作时机和写作基调上都有所不同：

[1]研究生（graduate student）本是指本科之后的高等教育，通常包括硕士研究生和博士研究生两种学位。现代汉语中“研究生”一词经常被滥用，用以单指硕士研究生，这其实是错误的。——译者注

形式 《博士研磨》是一本面向大众的回忆录，而非一本面向在读博士生的“成功指南”。尽管博士生也能在我的经历中学习到经验和教训，但我的目标并不是直接提供建议。对于博士生而言，市面上的“成功指南”和“建议专栏”已经不胜枚举，我也无意画蛇添足。这些文章充满了“持之以恒”和“不积跬步，无以至千里”等等空泛的词汇，但相反，回忆录的形式让我能丰富、具体地叙述发生在我自己身上的故事。

时机 《博士研磨》是我在完成博士学位之后马上着手写作的，而这正是撰写这样一本回忆录的最佳时机。不同于在读博士生的是，我可以在回忆录中对整个博士求学期间的工作进行系统的整理和反思；而相比资历较深的研究者而言，我可能更容易忠于攻读期间的经历，不会引入过多由研究经历带来的有选择性的观点和感受。

基调 尽管保持完全客观是不可能的，但我在写作《博士研磨》过程中仍然试图贯穿一种相对客观的基调。与其他作者不同的是，很多撰写博士相关文章、书籍，或者绘制漫画的人通常属于以下两类之一：

- 成功的教授或者科学家，他们通常给出一些冠冕堂皇的建议，比如他们可能会说：“研究生生活诚然辛苦，但它同时应该是一段美好的知识之旅，你应该享受这个过程、并从中受益.....因为我当年就是这样做的！”
- 或者是苦涩的博士研究生或者辍学博士，他们常常因为自己的经历留下了心理阴影，当提及博士生活的时候会用一种过分夸张、“看破一切”、自我怨恨的腔调：“啊，那时我的世界就是个活生生的地狱，我到底拿我的青春换了什么？！？”

冠冕堂皇的建议可能能激励一些学生，而大倒苦水的呻吟可能能引起另一些处境不佳的学生的共鸣，但作为大众读者而言，他们可能并不能感受到这些极端的情绪。

最后，在我开始讲述自己的故事之前，我希望强调一下，每个博士研究生的经历与他/她所在的学校、院系、研究领域、经费状况等都有巨大的关系。在我的读博生涯中，我感到自己非常幸运，能很大程度上自由自愿地完成自己的学业；我知道很多学生相比而言受到了很多的限制。我的故事只是一个孤立的数据点，所以我所呈现的故事可能并不能泛化、推广到每一个人。然而，我会尽力避免叙述变得过分局限于我个人的情况。

祝阅读愉快！

Philip Guo, 2012年6月

译者序

巫术与火刑

研磨的故事

献给所有热爱创造的人

前言

因为我本科的专业是电子工程和计算机科学，大部分本科同学在毕业之后，都马上投入到了工程性的工作中了。而我最终选择攻读博士学位，究其原因，一方面是受到来自父母潜移默化的影响，另一方面也和我在本科期间对工程性的实习产生的不良印象不无关系。

我父母从未要求过我去攻读博士学位，但我可以看出，终身大学教授[2]是他们最尊重的职业——而博士学位正是成为终身教授的必要条件。为什么终身教授会是他们心目中的理想职业呢？这其实并不是因为他们对纯粹的学术追求有什么不切实际的盲目推崇。尽管我的父母也都到过良好的教育，但他们同时也是非常现实的移民——一份终身教授的职位对他们而言，更大的吸引力在于终身制所带来的工作保障。

我父母的很多朋友都是在企业的工程性职位上供职的中国移民。由于他们在英语技巧和美国文化了解上的不足，他们中的大多数人在职业生涯中的经历都并不顺利，而这一问题往往随着年龄增长而更加突出。在假日聚会上，我经常能听到一个不变的主题：人们的工作因为难以相处的经理而处处不顺、成为年龄歧视和“玻璃天花板”效应[3]的牺牲品、甚至面临大

[2]在美国和加拿大等国家，终身职位（tenure）是指资深学者拥有的，除因正当理由外，免于被解雇的合同权利。与此不同的是任意性职位（at-will），规定劳动双方随时可以以任何理由终止劳动合同，不需要法定的正当理由，也不需要提前通知。大多数非学术研究类（工程性）的工作属于后者。——译者注

[3]玻璃天花板（Glass Ceiling）效应是一个政治术语，用于形容“在企业内升职过程中看

规模裁员和长期失业的风险。尽管我父亲不是一位工程师，而是在高技术产业就职，但他也难逃这种魔咒，在一系列和管理层极其官僚做派的斗争中失败，最终早早地结束了他在公司的工作。那时他还相对很年轻，只有45岁。

我的母亲则是这种不幸潮流中唯一的幸存者。她十分热爱自己在UCLA[4]作为社会学终身教授的工作。和她的大部分中国移民朋友不同的是，她享受终身职务保障，不用向老板汇报工作，可以几乎完全自由地追求她自己的学术兴趣，在她的研究领域也小有名气。亲眼目睹我母亲成功的职业轨迹，和父亲及他们的朋友们在职业生涯上的恶性循环两者之间的巨大反差，这在我高中和大学本科的学习生涯中留下了难以磨灭的印象。

当然，仅仅因为这种非理性的、年少时的恐惧就去选择读博显然是不明智的。为了让自己对在企业的工作生活有所印象，我在本科的每个假期都参加了工程性公司的实习。因为我工作过的办公室都碰巧只有我一个实习生，我被赋予了一种罕有的特权——我的工作职责是按照全职初级工程师的标准分配的。尽管在这个过程中我学到了很多技术上的技巧，我仍然觉得这种日复一日的工作极少需要思考且非常无聊；这也可能与我实习过的公司不是一流公司有关。我本科的很多朋友都在微软（Microsoft）和谷歌（Google）等一流公司实习过，并非常喜欢他们的实习经历，他们最终也往往在毕业后和这些公司签下了全职工作合同。

因为我对我的实习经历感到厌倦，而另一方面对本科时作教学助理（助教）和研究助理（助研）的经历比较感兴趣，我当时将未来的职业目标定在了大学教学和学术研究上。等到我在MIT[5]的第三年过半，我已经下定决心在毕业后攻读博士学位，因为这是实现我职业目标的必经之路。我

不到但难以逾越的障碍，通常见于少数人种和女性身上，且与这些人群的资历和成就并无关联”的现象。——译者注

[4]UCLA是加州大学洛杉矶分校（University of California, Los Angeles）的简称。——译者注

[5]MIT是麻省理工学院（Massachusetts Institute of Technology）的简称。——译者注

决定留在MIT，完成一个五年制本硕连读的项目，以此来在申请博士项目之前积累更多的研究经验，以期能被录取到更多排名顶尖的院系中去。

我找到了一位硕士论文的导师，并且就像很多壮志踌躇的年轻人一样，开始向他阐述自己不甚成熟、遑论完善的研究思路和计划。我的导师很耐心的听我说完了我的想法，但最终仍然成功地说服我进行一些和他的研究兴趣更契合、更主流的研究，而且更重要的是，这些研究项目更契合他的基金项目要求。因为当时我的硕士项目学费一部分来自我的导师从美国政府申请的一个研究基金，我有义务在基金所规定的范围内完成研究工作。因此，我听从了他的建议，并将接下来两年半的时间用于开发一类新的原型工具，用于分析由C和C++语言编写的计算机程序的运行时行为。[6]

虽然我并不是非常热衷于我硕士论文的项目，但事实证明选择一个和导师研究兴趣契合的项目是一个明智的选择：在他有力的指导下，我发表了两篇论文——一篇我被列为第一作者（主要作者），另一篇的位置稍微靠后——并且我的硕士论文获得了系里年度最佳毕业论文奖。这些成就，加上导师在我申请文档中的帮助，为我赢得了几所顶尖计算机科学系博士项目的录取。因为斯坦福是我的首选，在我收到录取的当天晚上，我甚至激动得几乎无法入睡。

我还非常幸运地得到了NSF[7]和NDSEG[8]研究生奖学金的垂青，这两者

[6]运行时行为分析（run-time behavior analysis）是计算机程序编写和维护中一种重要的技术，它可以帮助软件工程师更快、更准确地发现在程序编写时难以注意到的程序缺陷和漏洞（bug），进而完善其功能并提高稳定性。Bug一词的原意是虫子，现代被用于形容软件缺陷。这一说法一个可能的来源，是源于1947年电子管计算机刚刚问世前后。当时在哈佛大学的二号机（Mark II）中，一位操作员仔细排查后发现，二号机当时一些计算谬误的产生，是由于一只飞蛾被困在了一个继电器上。这件事被记录在了当时的实验室日志中，后来常被称为“第一次真正找到的bug（此处为虫子和程序缺陷的双关）”。——译者注

[7]NSF是美国国家自然基金会（National Science Foundation）的简称。NSF是支持美国各大高等院校进行基础自然科学研究的重要基金来源之一。该奖学金的申请只面向美国公民及永久居民开放。——译者注

[8]NDSEG是美国国防科学与工程研究生奖学金（National Defense Science & Engineer-

都只颁发给了大概5%的申请者。这两个奖学金为我免除了攻读博士的六年之中，五年的全部费用，也使我不必完成各种研究基金相关的研究项目。与我不同的是，在我的研究领域中，大部分博士研究生依赖于教授提供的研究经费和院系提供的助教经费。博士研究生的经费包括全额学费，以及大约每个月1,800美元的补贴，用于贴补生活开支。（在我的研究领域几乎没有人自费攻读博士，因为那样在经济上非常不划算。）

因为我已经有一定的研究和写作论文的经验，当我在2006年9月来到斯坦福时，我觉得自己已经为未来艰苦的博士研究做好了充足的准备。然而，那时我完全没有预见到的是，我的博士第一年即将成为我生命中到此为止最为打击信心、令人灰心丧气的一段时间。

---

ing Graduate Fellowship）的简称。该奖学金由美国国防部出资设立，旨在推动国防相关的科学与工程研究的发展，其申请只面向拥有美国国籍的人士开放。——译者注

第一年：碎梦大道

2006年的夏天，在我开始在斯坦福攻读博士学位的几个月前，我考虑了一些我认为自己感兴趣研究的课题。大题而言，我想要创造一些创新性的工具，用来帮助人们在进行计算机编程时提高效率，换言之，*提高程序员生产力*。我之所以对这个方向感兴趣，主要源于我在暑期实习中自己的编程经历：因为日复一日，公司分配给我的工作并不能让我提起太多兴趣，工作中的很多时间，我都坐在自己的格子间里反思，在我工作的这些公司中计算机编程的流程如何低效。那时我认为，如果能投身于旨在降低这种低效性的科学研究，应该是不错的方向。更宽泛地说，我的研究兴趣集中在能让所有类型的计算机使用者更加高效的工作中——而非仅仅聚焦在专业程序员的身上。举例来说，我希望能设计出新的工具，能帮助科学家分析和绘制数据、帮助系统管理员调整服务器配置、或者帮助计算机新手学习使用新的软件。

尽管我当时就有这些模糊、不成型的兴趣，但距离我将这些兴趣转化为真正可以发表的研究项目，并最终形成一篇*博士论文*，还有很多年的差距。对斯坦福计算机科学系的博士研究生而言，通常他们需要发表二到四篇第一作者的学术论文，并将这些论文合并成一篇博士论文——通常是一篇长度与书籍相仿的科技文档。当博士论文通过一个由三位教授组成的*博士论文委员会*批准后，学生就可以毕业，从而获得博士学位了。在我所在的计算机系，通常一个博士研究生需要四到八年毕业，这个年限取决于他们发表文章的效率。

在2006年9月的新生信息会[9]上，系里的教授鼓励所有的博士新生尽快找到自己的导师，所以我和我的同学们一样，把一开始的几个月花在了找教授谈话上，希望尽快找到一个研究方向匹配的导师。对于一个学生的论文委员会来说，导师扮演了最重要的角色，因为他/她对学生能否毕业拥有最终的决定权。在我的研究领域，导师通常还负责通过自己的科研经费为学生提供经费支持，并且指导他们开展课题研究、写作论文。和几位教授谈话后，我发现Dawson[10]和我的研究兴趣和研究风格都似乎最为接近，所以我选择了他作为我的导师。

在我刚去斯坦福的时候，Dawson已经在斯坦福度过了八年，并且刚刚获得终身教授职位；通常，教授在他们工作的前七年如果发表了足够多高水平的研究论文，就可以得到终身职位（终生的工作保障）。Dawson的主要研究兴趣是创造新的工具以自动在复杂、真实的软件中寻找bug（软件代码中的缺陷）。在过去的十年间，Dawson和他的学生编写了许多这样的工具，相比他们的竞争者而言，他们能找到程序中更多的bug。他们的研究成果十分有效——他们甚至城里了一个成功的创业公司，通过提供基于这类技术的缺陷查找服务而盈利。尽管我对Dawson的研究项目感兴趣，更吸引我的一点则是他的研究哲学和我自己的想法十分接近：他是充满激情的“务实派”——相比于单纯为了显得“学术”而去研究理论上“新颖”的课题，他更关注的是得到实实在在、有说服力的结果。

我和Dawson第一次面谈时，他似乎对我的大方向——让计算机的使用和计算机编程变得更高效——只是稍感兴趣。不过，有一点他说的很清楚：他非常希望招收一些新的研究生来帮助他完成一个叫Klee的软件缺陷

[9]信息会（Orientation），或译为迎新会，是在学校或大型组织中常见的，统一为新成员提供常用信息、帮助其尽快适应环境的活动，通常由资深成员主持。

[10]Dawson是这位教授的名字（first name），而非姓（last name），故翻译成Dawson教授是不准确的——原文也没有出现Professor Dawson（Dawson教授）这一称呼。单独使用名字在英语中非常普遍，通常用于熟人之间互相称呼、陌生人之间互相介绍、以及不希望提及全名的场合。文中出现的人名都是单独的名字。为了忠于原文的阅读体验，译者没有对这些名字进行标准化翻译和加工。——译者注

检查工具，因为这个项目是他现在的科研经费所支持的。（这个工具有好几个名字，但为了简便，这里就叫它“Klee”。）和其他的教授和高年级博士谈过之后我才意识到，对于新生而言，加入一个已有的、由研究基金支持的项目是一种常规现象，而并不是马上开始进行自己原创的研究项目。我说服了自己，认为自动查找软件缺陷也是一种间接提高程序员工作效率的研究，于是便打定主意，加入Klee项目组。

当2006年12月，我准备加入Klee项目组时，Dawson已经在指导五个学生参加这个项目了。项目组的带头人Cristi是第三年的博士生，而正是他和Dawson开发了最初版本的Klee。Dawson、Cristi和其他研究者不久前还发表了第一篇说明Klee系统的论文，并展示了Klee在发现一些新的缺陷上十分有效。那篇论文受到了学术界的好评，所以Dawson希望保持这一势头，继续发表几篇跟进这一项目的文章。值得注意的是，从同一个研究项目中发表多篇论文是可能的（即“跟进论文”），只要这些新的论文有新的原创观点，相比前作的改进和创新，或者相比前作而言在结果上有很大的改善。当时，下一个相关领域的顶级会议的论文提交截止日期是2007年3月，所以Klee团队有四个月用于基于之前的论文做出创新性的改进，以期发表一篇新论文。

~

在我继续讲述我的故事之前，我想简单地介绍一下学术论文是如何评审和发表的。在计算机科学领域，发表文章最受关注的场合是学术会议。当然，值得指出的是，再很多其他学科中，学术期刊才是最受关注的，而对这些领域而言，“学术会议”往往和计算机科学领域大相径庭。对计算机科学而言，学术会议的论文发表流程大概如下：

1. 每个会议发布一个征稿启事，其中说明了会议所要求的课题范围和一个论文提交的截止日期。

2. 研究者需要在指定的截止日期之前提交自己的论文。通常每个学术会议会收到100到300份论文稿件，每篇论文大概包含30到40页双倍行距[11]的文字。

3. 学术会议的程序委员会（Program Comittee，简称PC）通常由大约20位专家研究者组成，他们负责将论文负责分类，以便审稿。每篇论文通常由三到五个人完成评审，参与评审的人员可能包括PC的成员，或者由PC成员邀请的、来自学术界自愿参与审稿过程的审稿人。论文的评审过程通常需要大约三个月。

4. 当每位PC成员都完成审稿后，整个委员会将开会商议，通过审稿人的反馈决定接收一部分论文稿件，并拒收剩下的稿件。

5. 程序委员会会发出电子邮件通知所有的作者，告知他们论文是否被接收，并将审稿人对他们的论文提出的审稿评价附在电子邮件中。

6. 论文被接收的作者参加学术会议，并关于自己的论文做一个30分钟左右的演讲。学术会议结束后，所有的论文都将被收录在在线的数字图书馆中。[12]

通常，一个备受关注的顶级学术会议的论文接收率在8%到16%之间，而第二级的学术会议大概接收20%到30%的投稿。由于这些接收率相对较低，对于一篇学术论文来说，被拒收、修改并重新提交并不罕见——在论

[11]双倍行距排版时，行间距与文字高度相同，通常用于学术期刊初稿的排版。而计算机科学的学术论文通常采用单倍行距、双栏排版。——译者注

[12]根据研究领域不同，学术会议的举办方式略有不同。在一些领域中，论文会被PC分为演讲展示（oral presentation）和海报展示（poster presentation）两种形式。演讲展示中，通常有5至30分钟供作者在报告厅等场合公开真实自己的研究成果；海报展示时，所有参与的作者则将自己的研究成果展示在一张海报上，并以海报为基础向观众展示自己的研究。通常演讲展示会有更多听众和更大的影响力，在有演讲展示和海报展示之分的学术会议中，PC也会把有限的演讲展示机会分配给他们认为更有影响力的学术研究项目。——译者注

文最终被接收之前可能这一过程要重复多次，而这一过程可能会花费数年的时间。（在同一时间，一篇论文只能被提交到一个会议。）

~

当Dawson说他想要提交一篇论文到2007年3月截止提交的顶级会议之后，他向我介绍了当时其他五个学生工作的方向，并让我选择自己感兴趣的工作。我选择了使用Klee来寻找Linux*驱动程序*中存在的新的程序缺陷。*驱动程序*是指用于帮助操作系统完成其与外置设备（如鼠标或键盘）通信的软件代码。而Linux*操作系统*，和微软Windows系统或者苹果Mac OS系统类似，包含了成千上万这样的驱动程序，用以连接各种各样的外置设备。对于传统的调试方法而言，驱动程序中的软件缺陷十分难以找到，甚至有可能是危险的，因为驱动程序的缺陷可能会导致操作系统死机甚至崩溃。

Dawson认为Klee可以在Linux驱动程序的成千上万行代码中，找到其他自动缺陷分析软件（甚至人）从未找到过的软件缺陷。我记得我当时考虑过，尽管在Linux驱动程序中找到缺陷写在论文里看起来很不错，但我并不是很明白这样的工作能不能算是真正的研究贡献。按照我的理解，我要做的事情是用Klee去寻找程序缺陷——将一个已有的研究应用在实际问题上——而不是采用一种创新的方法来提高Klee的性能。此外，我并不明白，到三月份论文截稿时，我的工作和其他五个学生的工作如何能融合成一篇自洽的论文。尽管如此，我当时相信Dawson在头脑中有一个高瞻远瞩的思路完成这篇论文。介于我刚刚加入研究项目，我并不想马上开始对这些应该由教授决定的问题开始指手画脚。任务已经摆在眼前，我要考虑的只是尽我所能，完成这个任务。

~

我博士生涯的前四个月被我用于配置Klee以用它来分析上千行的Linux设备驱动程序，以期从中发现新的程序缺陷——这一过程并不

顺利。尽管看起来我的任务并不复杂，但我很快就被淹没在了一些细节问题中，而这些细节对于让Klee能够分析Linux驱动程序又是必不可少的。我经常会花几个小时设置好Klee所需的复杂的实验环境，以分析某一个驱动程序的程序缺陷；但这种工作又往往以Klee因为其自身的程序缺陷而崩溃告终，让我的努力付诸东流。当我把这些Klee中存在的问题报告给Cristi时，他会竭尽全力解决，但由于Klee本身极其复杂、环节众多，在其中找到并解决程序缺陷绝非易事。我并不是想要专门指责Klee：任何以科研为目的开发的原型软件都会或多或少有一些难以预见的缺陷。我的任务是用Klee去Linux驱动程序代码中寻找缺陷，但讽刺的是，我整个第一个月的工作都变成了在Klee里找缺陷。（Klee不能在自己的代码里自动找到缺陷，这实在是太遗憾了！）随着时间的流逝，我对于手头的工作越来越感到沮丧，我觉得自己被分配的任务就是单纯的苦力，毫无知识含量——我的时间都用在让Klee能正常运行上了。

这是我生命中第一次感受到被手头的工作所淹没的无望。以前，我的暑期实习都相对不难，而且就算一些学校的作业对我来说有些挑战，作业中也总有一个需要找到的正确答案等着我。如果我课上有没听懂的内容，助教或者高年级的学生总可以帮我答疑解惑。就算在本科进行学术研究的时候，我也总能请辅导我的博士学长帮忙，因为我当时处理的问题相对比较简单，而他通常知道问题的解决方法。对于本科的研究助理而言，对学术研究的投入以及期望也相对较低：科研只是我日常生活很小的一部分。如果我在某个科研问题上毫无头绪，我可以选择集中精力在课程作业上，或者干脆和朋友们出去玩。本科毕业也和学术研究毫无关联。然而，作为博士研究生，学术研究是我唯一的工作，除非我在学术研究上能有所成果，否则我将得不到博士学位。我难以把自己的情绪和每天的研究进展分离开来，而在那几个月中，我的研究进展出奇得缓慢。

我现在步入了一个完全陌生的领域，所以我很难再向本科时候那样向别人寻求帮助，因为问题的答案往往是不确定的。因为我是唯一试图将Klee应用于驱动程序代码的人，我的同事们并不能为我提供任何指

导。Dawson偶尔会给我一些较高层面的策略性的建议，但就像所有已经获得终身职位的教授一样，他的角色并不是和他的学生们一起“在战壕里（第一线）参与战斗”。在做出研究成果的过程中，弄清楚所有复杂的细节是我们学生的任务——对我来说，这些细节就是如何找到别人从未发现过的、Linux驱动程序中的软件缺陷。教授们都喜欢重复这句老生常谈：“如果有人曾经做过这样的工作的话，那就不能叫做学术研究啦！”这是我第一次亲身体会到了这句话的含义。

尽管我每天都觉得工作很无望，但我仍然不断安慰自己：*我只是刚刚开始在这里工作，我应该保持耐心*。我不想在我的导师或者同事面前显得软弱无能，尤其因为我当时是Dawson组里最年轻的学生。所以，我在超过100天的时间里，每天修复Klee不断产生的新问题，不断遇到新的、更加棘手的问题，在我的“寻找Linux驱动程序缺陷之旅”中，拖着沉重的脚步前进。

当时，在我醒着的每时每刻，我不是在工作，就是在思考研究中的问题，抑或是在因为自己在研究中被艰深的技术问题所困扰而感到沮丧。与一般的朝九晚五的工作（比如我的暑期实习）所不同的是，以前每天晚上我可以把工作留在办公室，坐在电视前放松自己；而现在的学术研究在情绪上和心理上都是无休无止的。我晚上几乎没法让大脑停止思考问题，尽情放松休息——后来我发现几乎所有的博士研究生都有类似的困扰。有时，因为我的研究任务繁重得难以想象，我甚至会因为压力而失眠。想要在研究中休息一段时间也几乎不可能，因为在3月份的论文提交截止日期之前，还有很多工作需要完成。

在做这些苦力时，我曾经试图想出一些半自动的方法，可以让每天的研磨不那么辛苦。我和Dawson交流过一些初步的想法，但我们最终的结论是，如果我们想让Klee能在Linux驱动程序里找到缺陷，这种耗时间的研磨是无法避免的。接下来，在论文提交前的几个月里，我只能继续辛苦工作。

理性而言，我十分理解在科学和工程领域中，通过实验式的研究方法获得研究成果，通常并不光鲜、甚至包含很多辛苦的工作。而博士研究生，尤其是第一、第二年的博士生，通常不得不在这一过程中参与最辛苦、最劳力的工作——这是他们的资金来源所决定的。通常，在一个研究组中，教授和高年级博士生制订高瞻远瞩的研究计划，并将任务布置给低年级的学生，由他们去研磨所有的细节，并让项目实际运转起来。第一、第二年的学生通常很难影响研究组项目的大方向。尽管我完全接受自己在这种“等级制度”下较低的地位，但感性上，我仍然深受打击，因为手头的工作实在是太他妈难了[13]，而且毫无成就感可言。

~

经过两个月的研磨之后，我收获了一些小小的成功。我成功地让Klee能在一些较小的驱动程序上良好工作，并开始找到了几个程序缺陷。为了确定这些缺陷是否真实存在（而非因为Klee的局限而导致的误报），我向开发这些驱动程序Linux的程序开发者发了几封电子邮件，说明了这些可能存在缺陷的地方。几位开发者回复了我邮件，确认了我的确是在他们的代码中找到了真正的缺陷。当我收到这些确认的邮件时十分激动，因为这是我第一次受到外界的认可，尽管它们只能算很小的鼓励。尽管我没有做出什么革新性的研究，我依然收获了一些成就感，因为这些缺陷在没有Klee的帮助下可能非常难以找到。

在这些小的驱动程序里的缺陷得到确认之后，我的士气有所恢复，所以我开始着手分析更大、更复杂的驱动程序。然而，接下来的几周中不断浮现的技术问题逐渐变得难以承受，并几乎把我推到了崩溃的边缘。这里简单总结一下我当时遇到的问题：Klee实际上只能在不超过大概3000行，由C语言写成的代码中找到缺陷。最小的Linux驱动程序代码大概有100行，所以Klee分析它们游刃有余。而稍微大一些的驱动程序就有大约1000行代码，而这些代码还与大约一万行到两万行的Linux操作系统

[13]原文此处为“so damn hard”，此处为保持原文语气，进行了直译。

代码密不可分。这就导致问题一下远远超过Klee可以分析解决的范围，因为Klee并不能把这1000行代码和其他部分“斩断联系”而单独分析。我试过很多方法减少这些外部的连接（术语称为依赖），但是这样做意味着对每一个驱动程序，我将需要花费几天的时间去完成十分复杂的工作，而最终这些工作也只对这一个驱动程序有效。

我和Dawson面谈了一次，向他说明了我面临的问题的复杂程度，也表达了自己对此的消极情绪。对我来说，在每一个新的驱动程序上都要花费几天的时间才能让Klee可以运行，这简直是荒唐至极的工作。这种工作不仅让我疲惫不堪，它本身甚至不能算是学术研究！在我们的论文里，我应该怎么描述我的工作——我花了大概1000小时的苦工，就为了让Klee能分析一堆驱动程序，除此之外我并没有其他可说的了？这根本不是学术贡献，反而听起来太愚蠢了。另一方面，我开始着急，因为距离论文提交截止只有五周了，而Dawson似乎还没有提过我们组的研究论文应该从什么思路着手写作。通常一篇优秀的学术论文需要至少四周时间才能完成，而且我们的情况可能更加复杂，因为我们有六个学生各自负责项目的一部分，我们需要把这些部分融合在一起。

我和Dawson面谈过后几天，他提出了一个改进的思路，让Klee能够克服我所面临的依赖问题。他提出的这个观点叫做约束下运行（UnderConstrained execution，简称UC），而这种方法也许能让Klee将Linux驱动程序和外部的一两万行代码剥离开来，并单独在驱动程序代码中进行分析。他马上就开始和一个高年级的学生开始将UC技术融合在Klee中；他们把改进后的版本称为Klee-UC。尽管我当时筋疲力竭、接近崩溃，但我仍然非常欣慰看到自己的痛苦挣扎至少激发了Dawson的灵感，想到了这个有朝一日可能成为学术贡献的新思路。

接下来，Dawson和那名学生在Klee-UC上花了几周的时间。与此同时，他们让我继续用原来的方法，手工在Linux驱动程序中寻找缺陷。他们打算通过用Klee-UC试图找到我用普通Klee找到的缺陷，来证明其有效性。他们想在论文里阐述的观点是，与其让一个博士生（就是我！）为了

找到每个缺陷而在繁杂的工作上花费几天的时间，不如用Klee-UC来在几分钟之内找到这些缺陷，而且不需要准备任何特定的实验环境。

在近乎疯狂地又研磨了几周之后，我终于让原版的Klee成功分析了937个Linux驱动程序，并找到了55个新的软件缺陷（其中32个收到了开发者的确认）。接着，我还得把刚刚成型的Klee-UC配置好，并准备分析这937个驱动程序。这比之前的配置还要麻烦，因为在我准备用Klee-UC分析驱动程序的同时，Dawson和那个学生仍然在（编程）实现它。谢天谢地，最终Klee-UC的确找到了我找到的缺陷中的大部分，这样我们做了一些学术贡献，也有成果可以在论文里展示了。

但是这里有个巨大的问题。等到我们得到那些不错的结果时，到论文截止日期已经只剩下三天了，而我们都完全没有着手开始写论文。在这么短的时间里，撰写、编辑、完善一篇文章，使得它能被顶级计算机科学学术会议录用是完全不可能的。但我们还是努力这么做了。在提交截止前的72小时里，Dawson和我们五个学生（其中一位现在已经退学了）两个通宵住在办公室，以期完成实验和论文。我们所有学生心里都知道这篇论文不可能被接收，但我们还是在Dawson的带领下，顺从地向目标前进。

最终，我们提交了一篇令人尴尬的文章，里面充满了错误拼写、不通顺的句子片段、没有解释的图表，甚至连结论段都没有。整个是一团乱麻。在那一刻，我难以想象，如果我的研究工作一直是这样毫无章法，我到底要如何从博士项目中毕业。三个月后，不出所料，我们收到的审稿意见极其消极，甚至包括这样的严厉谴责：“程序委员会（PC）认为这篇论文简直太过草率，根本不可能被录用；请不要提交这种甚至无法审阅的论文。”

~

这次惨痛经历之后，我申请了去谷歌实习，因为我急需换一换环境。那份实习和我的研究兴趣毫不相关，但我并不在意。当时我只想逃离斯坦福几个月。

这时是2007年4月份了，距离6月份我的实习开始，还有十周的时间。我不知道我能做些什么，但我想逃离我之前四个月折腾Klee和Linux驱动程序的经历，逃得越远越好。我根本不关心我们是不是会修改并且重新提交那篇论文（最后没有）；我只想逃离那段梦魇。但因为我已经积累了近千小时的使用Klee的经验，而这也是Dawson当时唯一关心的项目，我想这可能是一个开展新课题不错的切入点。于是，我找Dawson谈了谈，和他交流了一些“不寻常”的使用Klee的方法，而不仅仅是局限于寻找软件缺陷。

然而我很快就意识到，我完全没必要把自己局限在Klee上，因为我的经费来自NDSEG奖学金，而非Dawson的研究经费。而Dawson其他学生的境况则于我不同，他们除了继续在Klee项目中工作之外别无选择，因为他们都是靠Klee的研究经费支持的。所以我决定仍然让Dawson作为我的导师，但离开Klee项目组去从头开始开展我自己的研究项目。

为什么我不早点“单飞”呢？因为尽管我的奖学金理论上给予了我研究方向的自由，但我知道我仍然需要一位导师的支持才能最终毕业。当时，Dawson显然想让他所有的新生都把精力投入到Klee上，所以我就花了四个月的时间在Klee上研磨，塑造了一个“好士兵”的形象，而不是一开始就傲慢地要求做自己的项目。而且，就算我选择了别的导师，我一开始还是会需要花时间在他们的项目上，来证明自己的能力。这种“交学费”的工作根本无法避免。

接下来的十周里，我在一个完全真空的环境思考自己的研究思路，完全没有与人交流。因为我一开始对研究组留下了如此消极的印象，我现在只想自己一个人清静地思考。Dawson对我的消失并没什么意见，因为他并没有在用自己的研究经费支持我。

我把自己完全与世隔绝，心理上基本处于崩溃状态，但我任然试着每天能有所进步。我尝试着每天阅读几篇计算机科学的学术论文，并做好笔记，以帮助我激发自己的灵感。但因为没有良好的引导和研究背景，我最后浪费了很多时间，有时读完了论文却一无所获。我还曾经骑着自行车在

校园里漫无目的地游荡，试图在过程中思考自己的研究思路但常常无功而返。最终，我把时间都浪费在了拖延上，可能甚于我这辈子至今为止的任何一段时光：我看了很多电视节目，睡了很多觉，花了无数个钟头在网上蹉跎。与我的朝九晚五的朋友们不同的是，我没有老板每天监督我的工作，所以我可以无拘无束，让我的大脑自由思考。

尽管我对研究方向的头脑风暴大部分都毫无目标，但我的想法逐渐地开始向一个问题聚拢：*我们如何才能经验性地度量软件的质量呢*？这是在我开始读博之前，受到工程性实习中遇到的低质量的软件而启发，所想到过的一个感兴趣的研究方向。然而，在真空中空想研究思路的问题所在，是我当时没有足够所需的经验来把这些思路转化为现实的项目。因为我还没有准备好，显然这种完全的思想自由在当时更像是一种诅咒。

尽管我对创造新方法来度量软件质量非常感兴趣，我当时也明白这只能是一个模糊的梦，没有足够的形式化的研究方法支持，是不可能被学术界所接受的。如果我当时自己闭门造车研究这个项目，我的下场只能是成为一个胡说八道却又假装内行的人。我永远不可能有机会把这些想法发表在顶级、甚至次一级的学术会议中，而发表不了论文就意味着不能毕业。那时，我已经不再抱有虚无缥缈的梦想，想要成为一名终身教授了：我只想能找到个办法毕业。

在那整整十周的与世隔绝中，我几乎没跟任何人说话——甚至包括朋友和家人。找人抱怨也毫无意义，因为没人能明白我当时经历的一切。我的不读博的朋友们以为我就是“在学校上上课”。而少数几个我在系里新认识的朋友也和我一样，正在因为他们博士第一年里的挣扎感到抑郁——这些挣扎的主要部分，多半就是不得不直面充满挑战、不知从何入手的研究问题带来的震惊，以及无法影响实验室研究方向、只能随波逐流的无奈。我们这一群年轻的计算机科学家来到这里，自愿地投身于复杂而看起来毫无意义的工作中，所收获的却只是我们在公司工作的朋友们四分之一左右的工资。这种现实简直可悲到可笑的地步。但我觉得大家凑在一起互倒苦水也毫无裨益，所以我选择了闭嘴。我选择了不来计算机系的大楼里工

作，因为我害怕碰到同事、同学们。我害怕他们会不可避免地问我我现在研究的课题是什么，而我并没有什么可说的东西。藏身于图书馆或者咖啡店让我感觉更为安逸。

反观当初，这么早脱离实验室单干是一个非常糟糕的选择。和想象中一位孤独的学者坐在路边，一边喝着拿铁咖啡一边在一张笔记纸上涂涂画画这种浪漫化的场景不同的是，真正的学术研究从来就不是在“真空”中完成的。为了创新，一个人需要有坚实的知识、历史、甚至物质基础（比如实验室设备）。在那几周中，我应该追寻的更明智的路线是找更多机会和Dawson谈话，并积极寻求和其他教授以及高年级学生的合作。但当时的我太过崩溃和沮丧，不满于“等级分明”的基于研究组的学术风格——把新来的博士研究生放在最繁杂枯燥的工作中——所以产生了逆反而选择了单飞。

~

在我十周的与世隔绝接近尾声，即将去谷歌暑期实习之前，我给Dawson发了个电子邮件，简单说明了一下我最近读到的一篇让我思考了很多的科技博客。那篇博客让我想到了一个想法，那就是通过开发者在程序代码存在的整段时间里编辑代码的过程，来衡量一个软件项目的质量。令我惊喜的是，Dawson很快给我发了一个简短的回复，说他对这种衡量软件质量的技术也很感兴趣，尤其是如何用这种技术来帮助Klee这样的自动缺陷查找工具。

当知道Dawson对我感兴趣的领域也感兴趣之后，我再次燃起了希望，因为他也许能帮助我把这个想法变得更接近现实。我很快的为这个新想到的经验化软件质量度量的项目记下了一些笔记，并打算从暑期实习回来后开始研究这个课题。这样，在四个月与Klee的痛苦研磨，以及十周漫无目的的找寻方向之后，我博士生涯的第一年画上了一个还算乐观的休止符。

第二年：整装启程

在谷歌的暑期实习对我而言，是一段脱离学术研究的轻松时光。我的工作压力并不大，所以我经常有机会和其他实习生一起闲聊、交朋友。临近夏天结束时，我已经从第一年的痛苦和折磨中痊愈，做好了重新开始博士生活的准备。

在夏季最后几天，我给Dawson写了一封电子邮件，重申了我对追求自己研究兴趣的渴望，但同时强调了从研究中发表论文对我而言的重要性："我从这个暑期以及之前的暑期实习经验，发现自己很难集中精力完成一个博士科研项目，除非我对这个项目有强烈的归属感和科研的热情；所以我非常希望能在令我感到激动的研究方向，和学术界中的教授们广泛认为具有'研究价值'的问题中，找到一个交集，并为之奋斗。"

我准备继续和Dawson一起在经验化软件质量度量的课题上做一番研究，因为我们已经在我第一年结束的时候讨论过这一问题。然而，我当时预感这个项目可能会风险较大，因为这不是Dawson主要的研究兴趣，也不是他主要的专业知识的所在；对他而言，Klee仍然是最高优先级的工作。于是，我想到自己可以去寻找另一个课题，通过同时在两个课题上做研究（并寄希望于至少一个能成功），来冲淡风险。我将在这章稍后介绍我和Dawson合作的主要研究项目，但我想先说说我的另一个项目。

~

就在2007年9月，我准备开始我博士生涯的第二年之前，我去波士顿度假了一周，看望我大学时的朋友。因为我当时离得不远，所以我给几位我本科时就认识的MIT教授发了邮件，希望可以向他们寻求一些指导。他们和我见面的时候，大概都谈到了同样的事情：*要主动去找教授谈话，尽量去找一个共同的兴趣点作为研究课题；不论如何，把自己关起来闭门造车是万万不可取的*。这句简单的建议在我博士接下来的五年里反复得到印证，也最终指引我顺利完成了博士学习。

还在波士顿时，我就马上将这个建议印在心里并付诸行动。我给MIT计算机系一位叫Rob的教授发了一封*冷邮件*（向从未通信过的人冒昧发送的邮件），并礼貌地请求和他面谈一次。在这封邮件中，我简单地说明了自己是MIT毕业不久的学生，现在在斯坦福攻读博士，并希望能开发出一些让计算机程序开发者能提高工作效率的工具。因为我知道Rob也对这一研究领域有兴趣，我希望他能回复我的邮件，而不是把它标记为垃圾邮件。Rob非常慷慨地在他的办公室和我面谈了一个小时，期间我向他介绍了几个研究项目的想法，希望能从他那里得到反馈。他似乎对这些想法很感兴趣，这让我信心倍增，因为我的想法或多或少得到了一位研究领域内的教授的认可。可惜，因为我不再是MIT的学生，我没有机会和Rob一起开展研究。在我们面谈的最后，Rob推荐我和斯坦福计算机系一位名叫Scott的教授谈一谈，看看我有没有机会成功向他推销这些观点。

回到斯坦福后，我给Scott发了一封冷邮件，请他给我一个机会面谈。去面谈时，我针对想和他沟通的三个具体的研究思路准备了一些笔记，大致格式如下：

1. 问题的定义是什么？

2. 我准备用什么方法解决？

3. 用什么样的实验，才能有说服力地证明我的方法具有良好的效果？

Rob的一位博士学生，也是我的朋友，Greg，教会了我其中第三点——用实验指导思考——在提出研究项目思路中的重要性。教授们很乐意让自己的名字出现在发表的论文上，而计算机科学领域的学术论文通常需要强有力的实验支持才能有机会发表。因此，在提出项目的想法时就考虑到实验设计是十分重要的。

尽管我的介绍都没能赢得Scott的全力支持，但他仍然希望可以和我一起开展一个和我大致研究兴趣相关的研究课题。当时，他是一位刚刚来到斯坦福三年的助理教授（尚未获得终身职位），所以他非常希望能多多发表论文，早日拿到终身教职。因为我自己有奖学金，Scott并不需要用自己的经费支持我，所以对他来说也没有什么顾虑。

~

Scott所精通的，是一个称为人机交互（Human-Computer Interaction，简称HCI）的、较为贴近实用的子领域。和其它子领域不同的是，HCI的研究方法以人的需要为核心。一般来说，HCI的研究项目都是这样完成的：

1. 观察人，并弄清楚他们面临的真正问题是什么。
2. 设计一些创新性的工具，以帮助缓解这些问题带来的影响。
3. 用实验来评估这些工具，以检验它们是否真的可以帮助人们解决问题。

因为我希望创造能提高程序开发者工作效率的工具，Scott就先建议我去开发者们真正的工作场所，观察他们的工作，并借此发现他们真正面临的问题。具体来说，Scott非常好奇为什么现代的软件开发者使用多种编程语言，并且严重依赖于在网上搜索、并剪贴代码片段来完成开发。之前几十年的高效开发工具的研究中，都假设了开发者只用一种语言进行开发，

并且认为他们所处的环境中开发者的水平大致相同，而这些假设现在显然都早就过时了。通过观察现代程序开发者的日常活动，也许我就能设计出更贴合他们需求的新工具。

接到Scott给我的定下的大致目标之后，我就马上着手开始寻找可以让我观察的、专业的软件开发者。因为我刚刚在谷歌实习，我试过在那里找到目标，所以我发邮件给了我实习时的经理，他也同意将我的邮件转发给其他同事。我很快就收到了几封婉拒的邮件，因为没人想因为一个非雇员看着他们工作，而惹上知识产权的麻烦。接着我又给一些在斯坦福附近创业的朋友们发了邮件，但很快就发现他们和大公司的程序员一样，也收到种种限制。遗憾的是，他们更难以答应我的请求，因为为了避免竞争者抄袭，他们需要对自己的创业项目保密。而且，他们远比大公司的开发者们更加繁忙，所以也不太希望浪费自己的时间，去帮随便一个研究生完成这种对他们没什么帮助的科研项目。

我当时最后的希望是去谋智（Mozilla）观察软件开发者工作，因为谋智是一个非营利性的软件开发组织，他们的产品包括著名的火狐（Firefox）浏览器。因为谋智的软件项目都是开放源代码[14]的，所以我以为他们应该不怕会有一个局外人来观察他们的程序员工作。我没有试着发冷邮件（其实我都不知道应该要给谁发！），我决定直接开车去谋智的总部，直接登门造访。我冒然和我看到的第一个人打了招呼，并向他介绍了自己。他很热心地告诉了我两位谋智高层领导的电子邮箱，并说明这两位可能会对这样的研究合作感兴趣。我当天就给两个人发了冷邮件，令我感到惊喜的是，其中一位回复称对我的研究很感兴趣。不幸的是，那也是我收到的来自他的最后一封邮件；当我请求继续开展研究时，他就再也没有回复我了。

[14]开放源代码（Open Source），简称“开源”，是一种在自由软件开发者中非常流行的形式。开发者自愿将自己软件的源代码向其他开发者公开，而在一定的通用协议下（通常被称为“证书”），其他开发者可以基于这些软件代码进行改动和再次开发。开放源代码有助于鼓励开发者分享自己的成果，避免重复劳动。当然，尽管软件源代码是开放、可供人任意下载的，有些开源软件也是可以盈利的。——译者注

现在反观当时，对于自己试图在工作场所观察别人工作的尝试以失败告终，我并不感到奇怪。毕竟，我并不能为那些被我观察的专业程序员带来任何好处；相反，我只会干扰他们的日常工作。幸运的是，几年后，我有机会观察了一组（非专业）程序员——为科学研究而编写程序的研究生们——的工作，他们并不排斥我可能带来的小小打扰，反而非常乐于和我交流他们工作环境中的问题。这些访谈的经历，最终激发了我的灵感，并直接为我的毕业论文做出了很大贡献。

~

在寻找专业程序员的路线上碰了南墙，我就决定回头，开始在斯坦福内部寻找这样的机会了。当我看到一张海报，宣传系里即将举办一年一度的编程竞赛时，我马上给主办者发了一封冷邮件。我想他说明了自己想要观察学生们比赛过程的想法，他欣然接受了。

尽管学生的编程竞赛对说明现实中的软件开发环境而言毫无代表性，但总是聊胜于无。Scott的一个比我高一年的博士学生，Joel，也希望来参加这种观察。Joel和我把整个周六上午四个小时都花费在了观察几个学生参加比赛这件事情上。我们看到的结果很无聊，最终也没能从我们的笔记中整理出太多有用的东西。然而，这个经历促使我们产生了一个想法，那就是进行条件更加可控的*实验室研究*。

这时，Scott已经决定让Joel和我合作进行研究，一起开展一个实验室研究，而不是分散精力在不同的项目上。因为Joel和我的研究兴趣相似，Scott也可以通过让我们合作，减轻自己管理学生的压力。Joel主要负责了实验室研究的设计，我也同时乐得清闲，因为我同时还在和Dawson做另一个课题的研究。

Joel、Scott和我设计了一个时常两个半小时的实验室研究，研究中，一名斯坦福的学生将会被要求从头开始，编程完成一个简单的网络聊天应用。他们可以使用任何他们想用的资源，最主要的就包括在网上搜索现成

的代码，或者已有的教程。我们找到了20名学生参加实验，其中大部分来自Scott当时在讲授的*人机交互导论*课程的学生中。

在接下来的几周里，Joel和我在系里地下室的机房坐了50个小时（20个学生，每人两个半小时），观察参与研究的学生，并同时将学生使用的电脑屏幕录制下来。一开始，观察学生如何工作非常引人入胜，但好景不长，我们很快就发现这个工作变得冗长繁杂，而且总能看到学生们一遍又一遍地做同样的操作、犯同样的错误。之后，我们又大概花了50个小时重放我们录下的视频，并在视频中标记了一些关键事件发生的时间。最终，我们分析自己的笔记，并回放标记好的关键事件，得到了一些关于“当这些学生面对一个简单、但实际工作中可能遇到的任务时，会遇到哪些问题”的深入了解。

我们把实验室研究的成果写成了论文，并将论文提交到了一个顶级的HCI学术会议。三个月后，我们发现论文被录用了，审稿意见非常令人振奋，而论文甚至还被提名了*最佳论文奖*。在这篇论文里，Joel是*第一作者*，而我是第二作者。几乎所有计算机科学领域的论文都是由多位作者合作完成的，而作者的顺序也是至关重要的。名字出现在首位的作者通常是研究项目的负责人（比如Joel），他通常在项目中完成的部分远多于其他位置靠后的作者，所以也最应当受到承认。其他的作者通常都是项目中的助理——通常是年轻的学生（比如我）或者远程的合作者——这些人也做出了足够多的贡献，可以被列为作者之一。博士生们通常把自己的导师（比如Scott）列在最后，因为导师通常会在研究想法形成、项目规划和论文写作方面提供帮助。

因为我不是论文的第一作者，这篇论文也就对我的毕业论文没有什么帮助；然而，这个经历教会了我许多关于如何开展研究、以及如何写作学术论文的知识。更重要的是，看到这个研究项目最终发表了一篇顶级会议的论文，我感到很有成就感——这和我博士第一年被拒收的Klee的论文形成了鲜明的对比。

Joel后来继续沿着这条研究路线走了下去，把我们的论文转化成了他博士论文的第一部分。在接下来的几年中，针对在一开始的实验室研究中发现的程序开发者们面临的困难，他又创造了几个能帮助克服这些困难的工具，并发表了论文描述这些工具的原理。同时，Scott也没有积极招我入麾下，我也就没有考虑过更换导师的事。当时看来，似乎我和Joel的兴趣太过接近了，而且Joel已经在他的子领域小有成就了。所以，我继续将研究重点放在和Dawson一起研究的课题上，毕竟他仍然是我的导师。然而在这一边，研究进展就没这么顺利了。

~

读者可以回忆一下，在我博士第一年接近尾声时，我开始在一个被称为*经验化软件质量衡量*的子领域寻找研究灵感——具体来说，就是通过软件开发的历史记录，来衡量软件的质量。正好Dawson也对这一课题感兴趣，所以在我博士第二年里，我们就继续在这一问题上开展研究。他的主要兴趣还是在于发明像Klee这样的自动缺陷查找工具，但他也在软件质量方面有一些兴趣。对他而言，让他感兴趣的研究课题包括：

- 如果给定一个大型软件项目，其中包含上千万行代码，如何区分哪些部分对项目而言更加重要，而哪些部分不那么重要?
- 有哪些因素影响了一些部分更容易出现软件缺陷? 比如，是最近刚刚被新手修改过的程序代码更容易包含更多缺陷呢，还是短时间内被很多人修改过的程序更容易出问题?
- 如果一个自动查找缺陷的工具找到了1000个可能的问题，哪些会是其中更重要的呢? 程序开发者可能并没有时间和经历处理全部1000个缺陷，所以他们必须根据一个重要度的评估，来优先完成重要的部分。

我通过分析和Linux系统核心这个软件项目相关的数据集，开始调查这些问题。我选择了Linux作为研究对象，因为它是当时最大且影响最广

泛的开源软件项目，有数以万计的程序开发者在二十年间贡献了几千万行的程序代码。Linux完整的*版本控制历史*能在网上免费得到，所以这成为了我主要的数据来源。项目的版本控制历史记录了在项目的整个生存周期中，所有代码文件中出现过的所有改动，包括改动出现的时间，更重要的是，改动是谁做出的。为了将这些项目历史和软件缺陷对应起来，我从Dawson的软件缺陷查找公司拿到了一份数据集，其中包含了公司的一个产品在Linux中发现的2000个缺陷。当然，这不会是Linux里所有的缺陷，但这是我能接触到的数据集中，唯一能为我提供了足够的信息，来研究我们的学术问题的了。

那时，我每天的工作流程就是写程序来从Linux的版本控制历史以及那2000个缺陷报告中提取、和清洗数据[15]，并将它们转化成便于分析的格式，并进行分析。为了能对这些数据有更深层次的认识，我自学了量化数据分析、统计学和一些数据可视化的技巧。在研究过程中，我细致地将自己的实验进展记录在了一本*研究笔记*中，用来记录哪些尝试并没有给我想要的结果。大概每周，我都会和Dawson面谈一次，汇报自己的研究进展。我们的会面通常是我想他展示我从分析中得到的图表或者表格，而他则会提供一些较高层次的评价，比如："嘿，这张图的这个部分看起来不太对劲。这是为什么呢？试试这样分割你的数据，并做一些更深入的分析。"很多年后，我意识到这其实是一种对于很多学术领域中的*计算研究者们*而言，再平常不过的工作流程；而在我的博士论文中，我也创造了一些工具，来消除这一过程中的一些效率不高的环节。不过那时我还没有这么长远的计划；我所想的只是做出一些有趣的发现，并把他们写成论文发表。

~

[15]数据清洗（Data cleansing）在科研项目中通常被广泛应用，尤其是当数据来自实际的项目，或数据产生、采集的过程没有严格监控时。数据清洗通常包括丢弃数据中对研究没有帮助的部分，或者丢弃因格式、内容、完整性等问题不能被用于研究的"脏数据"。此处作者应该指前者，即将和研究相关的数据分离出来的过程。——译者注

Dawson和我在将我们的成果发表的道路上并非一帆风顺。在那一年里，我们提交了两篇论文，最后都以被拒收场。这个研究项目后来又经过了一年，才转化成为一篇长度较短的、第二级学术会议中的论文得以发表，论文影响力微乎其微，也没有“作数”成为我博士论文的一部分。不过到那时，我已经不太计较这些了，因为我已经在开展其他研究项目了。

我们的论文在发表上遇到了这么多障碍，究其原因是因为我们不是经验化软件质量衡量（有时也被称为经验化软件工程）这一子领域的“圈内人”，而我们的论文恰恰属于这一子领域。在我们步入这个方向之前，已经有几十所大学的研究小组在进行类似的研究了。Dawson和我在这种竞争中完全处于劣势，因为我们的竞争对手包括很多专于此道的教授和科学家，而他们手下通常有很多博士生为他们产生研究成果。这些人对发表论文如饥似渴，因为他们中很多人仍然是年轻的教授，渴望借此得到终身职位。他们也更擅长于一些能让论文更容易被接收发表的技巧，包括统计方法、相关文献的引用，以及“营销技巧”等。最重要的是，他们中很多人在相关的学术会议中担当了外部审稿人或者程序委员会成员等职务，所以他们很清楚如果一篇论文想要被这个领域的学术会议所接收发表，应该需要怎么撰写。

读者可以回忆一下，我介绍过每篇论文都会由三到五位自愿参与的领域内专家——通常都是领域内的教授或者科研工作者——进行同伴评审[16]来决定是否可以发表。如果审稿人认为文章值得发表，文章就会被接收并发表；否则，作者就需要修改论文，之后再重新投稿。同伴评审的目的在于保证所有被发表的论文都有一定的质量保障，这一保障的水平则由

[16]同伴评审（Peer-review）是一种广泛用于计算机科学学术会议的评审模式，其中参与审稿的审稿人可能是领域内的教授、专家、研究者或在读博士生（本科生和硕士生较为罕见），这些人可能也是学术会议中的投稿人（审稿人不能审阅和自己有利益相关性的——比如自己撰写的——稿件）。“同伴”一词即用于强调这种审稿人和投稿人之间的平等性。与此不同的是，很多学术期刊采用专业审稿人审稿的制度，审稿人往往是领域内名至实归的教授或者科研工作者，一般认为他们在领域内的经验、学识等比一般投稿作者都略高一筹。——译者注

学术圈来决定。这种检阅的机制是必要的，因为总要有一些公认的标准，来过滤掉一些不靠谱的言论。然而，同伴评审这种制度本身存在其缺陷，尽管它努力使得评审结果更加客观，审稿人也仍然是人，还是不可避免地有自己的品味和偏见。

因为学术会议通常只接受不超过百分之二十的论文，如果一篇论文给审稿人以不好的第一印象，可能就会被拒收。Dawson和我都不是经验化软件度量领域的专家，所以我们并不能以一种审稿人所期待的方式，去“推销”自己的论文。所以，我们经常因为一些负面的评审意见受到打击，比如：“结果，我发现我就是对作者展示的结果不太信任。我的疑惑来自于两个方面：一方面我不相信他们对度量方法的认知是否合理；另一方面他们也没有使用一些有效的统计方法验证自己的成果。”在发表学术论文这个残酷的世界中，仅仅对一个课题感兴趣远远不足以让自己的论文得以成功发表；想要自己的论文可以发表，就得非常清楚作为论文审稿人的、自己领域中年长的同事们的偏好。简而言之，我们的数据集没有别人的好，技巧没有别人的细致，我们的成果和展示方式也和这一领域中有经验的研究者所期待的形式似乎格格不入。

相比而言，我和Scott、Joel发表的论文就更加成功，因为Scott是HCI领域的专家，并且在这一领域（尤其是我们发表文章的学术会议中）已经发表和评审了很多论文。当然，作为内行人并不能让审稿人在评审我们的论文时，对标准有任何的放松，因为那就太不公平了。但是，Scott仍然可以利用自己的经验，让我们的项目从动机到成果，都以一种审稿人所期待的形式展现出来，从而论文被接受的概率也大大上升。

~

在我博士第二年接近尾声时（2008年6月），我因为仍然缺乏有说服力的成果，又看到在经验化软件度量领域的论文如雨后春笋般地冒出来，开始变得越来越沮丧。我仍然没能将自己的发现发表成论文，也意识到自己和圈内的老手根本没法竞争：因为Dawson和我都不知道审稿人期待看到

的是什么，我开始明白继续在这个领域试图发表论文，只能是一路逆水行舟。又因为发表论文是毕业的先决条件，我必须尽快找到一个新的项目，否则我就不能拿到自己的博士学位了。在我准备在斯坦福开始自己博士生涯的第三年时，我非常绝望，以致不愿放过任何一棵能为我带来论文发表的救命稻草。也就是那时，我选择了回到博士第一年曾经让我痛苦不堪的项目——Klee——当中去。

第三年：噩梦未止

插曲

第四年：卷土重来

第五年：峥嵘岁月

第六年：尘埃落定

结语

后记