**在得到大学，如何做一个好选题？**

亲爱的同学，欢迎你加入得到大学。

欢迎你来报选题，参加分享前的打磨会。

开始填写之前，你还可以问问自己这些【启发式问题】：

1. 你们这行最容易被误解的地方是什么？遇到外行你最想替自己这行解释清楚的是什么？

2. 一个刚入行的人，刚开始最容易忽略，但是后来对他影响非常大的坑是什么？

3. 在你所在的行业中，绝顶高手和一般高手最大的区别是什么？

4. 到目前为止，你做过的最自豪，最有成就感的事儿是什么？（最好来自你的工作）

5. 如果有一天你转行了，你在现在这个行业里的积累，有哪些是可以马上带走的能力？

是不是有点儿感觉了？如果有感觉，可以试着填下面这个选题单：

|  |  |
| --- | --- |
| 选题单 | |
| 你的姓名和班级： | 陈文经 深圳2班 |
| 你的职业和身份标签： | 程序员 |
| 你想分享的主题是什么：  （上述五个问题，你想回答哪一个？请把你的回答填到这里来） | 如何让”错误“（bug，缺陷）从软件开发的不愉快的结果变成软件开发的驱动因素。这些经验对非程序员有什么启发。 |
| 你想分享内容的标题是： | 怎样让错误驱动进步 |
| 为什么你特别有资格分享这个主题（也就是：你比别人牛的地方在哪儿） | 作为程序员，几乎每天都在“写bug”。 |
| 关于这个主题，你有什么看法和做法  跟其他人是不一样的？ | 1. 在软件开发过程中，与其将时间花在后期来解决错误，不如让错误在开发的早期就暴露出来，从而让错误成为软件开发的驱动因素。   2. 找一切机会，缩短错误的反馈周期。 |
| 你准备从哪几个点来论述这件事情？  （列出你认为的核心点） | 1. 通常的软件开发流程，是从需求到程序员实现，再到测试。测试找出bug再返工给程序员。（这其中存在的问题） 2. 软件和bug的特点：软件之间相互关联紧密。往往定位很花时间，但解决起来比较容易。 3. 更高效但反直觉的开发过程是测试驱动开发。程序员自己（而不是测试人员）先编写好测试用例，然后再编写代码，让代码通过测试。（这样做的好处） 4. 为什么测试驱动开发这么好？结果却很少有团队采用？ |
| 对于一个听众来说，如果认真的听了你的分享，会对他有什么用？（听众听不听你这个分享会有什么不同） | 1. 如何让自己和团队更容易接受错误。 2. 如何让错误成为驱动自己成长的动力。 |
| 还有，所有的道理都需要故事和案例来支撑，尤其是你解决这个问题过程中的真实案例。所以来见你的教练前，请把你的故事和案例准备好。 | 1. 从一次深夜发布，第二天上午又紧急发布的故事开始。 2. 每个bug 2.56元红包的故事 3. 驱动测试开发为什么难以执行的几个故事 |

上次班班说到以一个让专业人士觉得厉害的案例贯穿分享，我思前想后，可能受限于自己的能力和工种（不是做后台开发，业务逻辑相对比较简单），确实没能找到合适的案例即适合外行也适合适合行业内程序员的例子。我想与其找一个NB的案例，介绍测试驱动开发这种适用面比较广但采用率低的方案，对外行更有帮助。

最后温馨提醒一下还要注意两点：

1. 内容可交付：即便不是同行，也能听懂你说的是什么。
2. 知识可迁移：不同领域的人听到后，对看待和处理自己的挑战，会有启发。

填完表，是时候联系你的班主任和打磨教练啦：-）