《个体软件过程》读后感

SY2206121 王一鸥

我阅读的书是Watts S. Humphrey的《个体软件过程》。本书介绍的方法——个体软件过程（PSP），是一个用以帮助软件工程师测量和改进工作方式的框架。PSP将介绍如何指定计划并跟踪个人的工作，并指出怎样始终如一地生产高质量的软件产品。

本书共20个章节，可以分为三部分，首先第一章是对软件工程的一些总述的内容，第二章至第十章关注软件项目的时间管理，第十一章至第二十章关注软件项目的质量管理。最后一部分与我们的课程更加贴合，所以也是重点阅读的内容。

本书的后半部分——第十一章至第二十章，讲述软件质量。我总结起来主要包括两个主题，一是关于缺陷的，包括缺陷的度量、查找、排除与预测等；二是论述使用PSP会提升软件质量，并减少缺陷排除成本。

软件现已应用在各个场景中。有些场景，软件一旦出现缺陷，会导致大量财产损失，甚至导致人身伤亡。所以软件的质量至关重要。软件工程师应该从内心认为软件质量是一件很重要的事。软件工程师需要对软件质量做出承诺，这样可以让他更加关注质量问题，认为生产出无缺陷的程序是很有成就感的。要将生产高质量软件视为一种自我价值的实现，这比外在的钱和权更加重要。

软件质量体现在许多方面，但首先要面对的而且必须解决的方面是软件缺陷。所以，书中首先对软件缺陷展开论述。**缺陷**，即指程序中存在的错误，例如语法错误、拼写错误、逻辑错误等。缺陷可能出现在程序中和设计中，甚至在需求、规格说明或其他文档中。缺陷是可以分类的，书中将其分为10大类。为了认识缺陷，需要收集个人的缺陷数据：记录发现的缺陷，为其归类、统计，设计发现和修复这类缺陷的方法。书中提供了”缺陷记录日志“表，用来记录收集的缺陷数据。

缺陷查找，主要通过四种方法：个人复查程序、依靠编译器、依靠测试、依靠用户反馈。其中”个人复查程序“是最快和最有效的，广义上也包含工程师彼此复查程序，即同行评审；编译器只能找出语法错误，不能找出逻辑错误；测试非常耗时且无法覆盖全部的可能；用户反馈后再查找和修复缺点，是最耗时的。在后面的章节中也可以看出，作者非常鼓励在编译之前进行缺陷查找和排除。书中也提供了”代码检查复查表“，来记录和统计个人复查程序时发现的缺陷。

缺陷预测，根据个人历史数据来预测本次项目各阶段的缺陷数。使用历史缺陷密度乘以本次项目预估代码行数来计算得到。我认为该预测的用途是在计划阶段，通过预估会出现的缺陷数，来为缺陷的查找和排除分配相应时间。

接下来，作者从不同角度论证，**在编译之前排除缺陷**，是有益的。从**经济效益**来讲，缺陷排除消耗人力、金钱。在越靠后的阶段，缺陷排除效益越低，每小时排除缺陷数也越低。所以越依赖编译、测试阶段的缺陷排除，其时间成本越高；在编译前进行缺陷排除，能够减小时间成本。从**产品质量**来讲，在编译前排除缺陷，会使产品质量更高。将审查、编译、测试中的缺陷排除过程比作一个个过滤器，过滤器按照比例来过滤缺陷，没有过滤率为100%的过滤器。如果输入的程序质量低，那过滤器输出的程序质量也会低。所以，要提高最后一个过滤器输出的程序的质量，在编译之前就要好好检查程序。从**过程质量**来讲，通过质检/过失比（A/FR）来度量第一次编译前花在查找缺陷上的时间的相对值，其值应大于2，才能保证较低的缺陷数和较高的质量。

阅读本书，对我有一些启发。首先，收集数据很重要，只有通过数据才能知道质量改善的效果，知道下一步如何更好地改善。书中提供了许多用来收集数据的表格，并详细讲述了使用方法，可以自己实践起来。然后，书中提出个人对代码的检查，比编译器检查、测试用例检查要更重要。这点确实和我以前的认知不一样，可以在今后的开发过程中，练习在编译之前自己检查代码。最后，本书提供的软件质量方法虽然较为简单、基础，但每一步都很详细，让我对软件质量管理有了更加具体的理解。

本书后续还有一本《团队软件过程》，后续有时间会继续阅读学习。