ch3习题12：

根据我们组内部的情况进行题目涉及事项的讨论，得到以下结论：

1）使用不同的语言设计项目会使整个工程代码的行数产生巨大差异，首先明确一个项目是以多个文件构成，其中有相当一部分为代码文件，我们这里称这样构成项目的小型代码文件为节点文件，整个项目便是由多个节点文件交互组成。而每个节点文件使用不同的编程语言，可能各文件的代码行数差距不大（但不排除会有极个别节点文件更换语言后代码量出现较大的差异），但是组合后整个项目在不同的语言编程后的代码行总量就会体现出明显差异。并且不同的语言的功能不同，比如python中就有很多C语言中没有的集成好各种功能的库等，这也会一定程度地影响到代码量大小。因此，我们认为仅用代码行数作为生产率的测度是不准确且波动性较大的。

2）对于人才招聘系统这一个项目的实现是一个较熟悉的数据库调用程序设计，但是其实现会有许多未知点，这会导致我们在编写程序时并不能按照之前的编程速度进行估计，也就是说不同实现的难度不同，所以即使是同一个人写代码的速度也会有所浮动，不能完全依照已有经验估计。因此，凭借成员过去的编程经验（写代码速度为主）进行生产率测度的估计并不适合。

3）如果项目提出者规定了不切实际的生产率作为工作量标准，那么我们会出现为了完成任务而对项目进行应付，编写冗余或无用的代码，或者把一行代码拆开写来打到行数标准。因此生产率的估计一定要切合开发成员的实际能力。

另外，为了更准确地测量生产率，可以考虑使用其他指标，例如完成的功能点数、软件的用户满意度、项目的进度等，或者综合考虑多个指标。同时，应该确保开发团队使用最佳实践，以确保代码的质量和可维护性。

ch3习题11：

学生软件开发项目可能面临的风险包括但不限于：

·时间管理风险：项目的时间表可能会因为学业或个人问题而受到影响，导致无法按时完成项目。

·技术风险：学生可能没有足够的技术知识或经验来完成项目，或者选择了不适合的技术工具或平台。

·范围膨胀风险：学生可能会尝试实现过于庞大或过于复杂的项目，导致无法按时完成或无法达到项目目标。

·人员风险：如果项目涉及多个学生，可能会出现一些人员方面的问题，例如缺乏沟通或协作能力，或者出现个人问题导致无法参与项目。

·质量风险：学生可能无法保证项目的质量，例如代码质量、测试覆盖率和文档完整性等方面存在缺陷。

风险暴露是指风险成为实际问题的可能性。

为了减轻学生软件开发项目的各种风险，可以采用以下技术：

时间管理技术：通过制定详细的时间表，利用时间管理工具和技术，确保项目按时完成。

技术管理技术：使用合适的技术和工具，对技术风险进行评估和管理，例如选择合适的编程语言、框架和开发工具。

范围管理技术：通过定义清晰的项目目标和范围，确保项目的范围适当，并能够在规定的时间内完成。

团队管理技术：通过管理团队成员之间的沟通和协作，确保项目团队在整个开发过程中保持高效运作。

质量管理技术：通过代码审查、测试和文档化等质量管理措施，确保项目达到高质量标准。

另外，一部分宏观的风险处理方法也能帮助风险管理：

1）规避风险：如果能够在项目早期就识别出风险，则可以采取措施进行规避，规避不了的，则考虑减弱影响；

2）接受风险：有些风险虽然存在，但是即使发生了，其影响也在项目可承担范围内，在考虑项目时间进度和成本的前提下，这类风险可以选择接受；

3）减弱风险：对于已知的潜在风险，并且不可规避的，可以提前策划，采取积极措施，减弱风险发生时对项目的影响，暂时无法减弱风险的，则应该持续监控；

4）消除风险：如果风险可能会带来灾难性的后果，则必须采取措施进行消除，如果不能全部消除，那么应该想办法减弱其影响，使其变为可接受范围，或者可规避，否则应该考虑调整项目的实现途径或者取消项目；

5）转移风险：风险一般都是带有特定背景属性的，一个项目组的风险，对于其他人也许就未必了，因此要运用宏观思维，借助外部力量，转移风险。