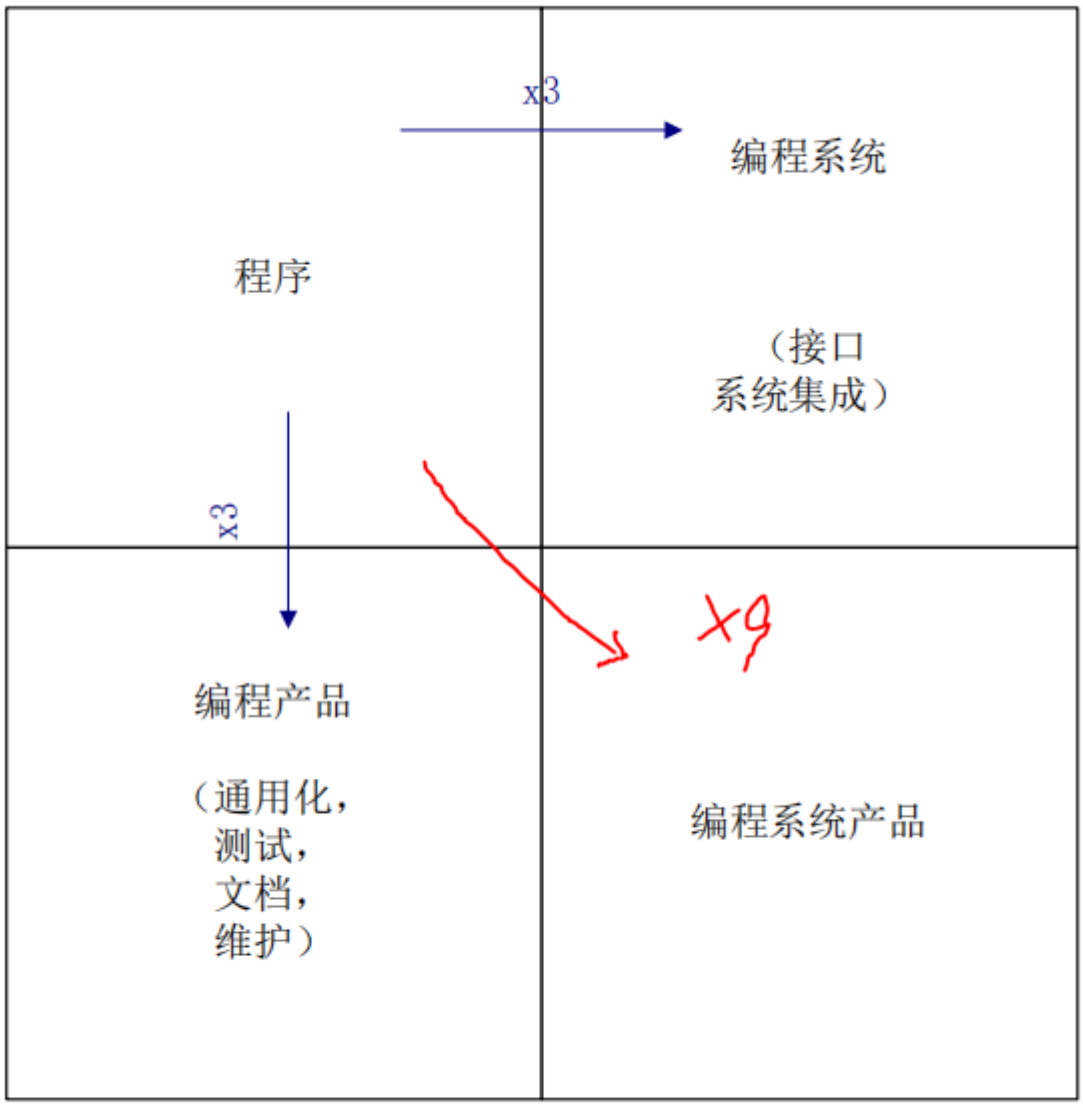
出处

《人月神话》

> 博主-天道酬勤-根据自己的工作经验摘录了自己认为较为重要的知识点汇总成此博文。仅供自己和大家参考！

一、知识点汇总

1、编程系统产品



编程系统产品的演进

- 编程产品
  - 可以被任何人运行、测试、修复和扩展的程序。

    - 多系统通用，程序符合编码规范。
- 编程系统
    - 作为一个结构单元符合规范，符合内存空间、输入输出设备、计算机时间的资源限制。

2、软件任务的进度安排经验法则

（1）任务进度安排经验法则

- 1/3计划
- 1/6编码
- 1/4构件测试和早期系统测试
- 1/4系统测试，所有的构建已经完成

  项目的延迟提交会付出高昂的商业代价！

  在众多软件项目中，缺乏合理的时间进度是造成项目滞后的最主要原因，比其他所有因素加起来的影响都大。

3、编程项目的组织架构

（1）开发成本

- 要协作沟通的人员的数量影响着开发成本，因为成本的主要组成部分是相互的沟通和交流，以及更正沟通不当所引起的不良结果（系统调试）。
- 减少交流的方法是人力划分和限定职责范围。

（2）职能介绍

- 产品负责人
  组建团队，划分工作及制订进度表，调用必要的资源。
  主要的工作是与团队外部，向上和水平地沟通。
  建立团队内部的沟通和报告方式。
  确保进度目标的实现，根据环境的变化调整资源和团队的架构。
- 技术主管
  对设计进行构思，指明产品外部呈现形态，勾画内部架构。
  提供整个设计的一致性和概念完整性。
  提供技术问题的解决方案，根据需求调整设计。
- 产品负责人和技术主管是同一人
  一般在10人的小型团队中存在。大型项目（**程序员超过25人**）中一般不出现，原因有二，一是同时具备管理技能和技术技能的人很难找到，二是工作量大，个人精力有限，难以应付。
- 产品负责人作为总指挥，技术主管充当其左右手
  该种情况下，技术主管不参与管理工作，这会导致技术主管很难建立技术决策上的权威。（决策权来源于管理）
  大的项目中这样的安排较为合适。
- 技术主管作为总指挥，产品负责人充当其左右手
  这样的安排能使工作非常高效。对中小公司一般项目较为适合。

4、程序的开销

（1）作为成本的程序空间

程序的运行，除了时间以外，所占据的空间也是主要开销。比如老款笔记本2G内存的，需要加装内存条才能应付win10系统所占的空间开销。

（2）规模控制

项目经理需要在技术工作和管理工作两个方面做规模控制。 确切定义模块的功能，并对模块规模（时间、空间开销，代码与库自身的文件大小）做限定。 功能模块过度优化影响系统的完整性。

（3）空间技能

将功能分解到很小的模块会耗费空间和降低性能。 一是采用功能交换尺寸， 二是考虑空间-时间的折衷，（1）培训团队编程技能（2）开发公共单元构件[每个项目要有能用于队列、搜索和排序的例程或者宏库]。对于每项功能，库至少应该有两个程序实现。

（4）数据的表现形式是编程的根本

程序的核心是：数据或表的重新表达。

5、文档先行

- 文档的跟踪维护是项目监督和预警的机制。
- 软件项目文档
  做什么：目标。定义了待完成的目标、迫切需要的资源、约束和优先级。
  做什么：产品技术说明。以建议书开始，以用户手册和内部文档结束。速度和空间说明是关键的部分。
  时间：进度表
  资金：预算
  地点：工作空间分配
  人员：组织图
- 正式文档
  （1）书面记录决策是必须的。这能确保分歧明朗，矛盾突出，各种细小的决定能得以保存，呈现出清晰、确定的策略。
  （2）文档便于同事间沟通
  （3）便于判断项目所处的状态

6、软件产品成功的保证--软件的设计和测试

（1）剔除bug的设计

- 防范bug的定义
  产品的正确定义是产品成功的关键。
  细致的功能定义、详细的规格说明、规范化的功能描述说明以及这些方法的实施，大大减少了系统中必须查找的bug数量。
- 测试规格说明
  在编写任何代码之前，规格说明必须提交给测试小组，以详细地检查说明的完整性和明确性。（程序员一般按自己理解的去做）
- 自顶向下的设计
  清晰的结构和表达方式更容易对需求和模块功能进行精确的描述。模块分割和模块独立性避免了系统级的

bug。

（2）结构单元调试

- 注意测试用例

（3）系统集成调试

- 系统调试花费的时间会比预料的更长

7、软件产品成功的保证--任务执行约束

（1）文档化

制定清晰、完整、准确的规格说明书

（2）会议

半天周例会，各执行负责人参与，并作会议纪要，记录各项决定。调整项目进度，修正项目前进方向。

（3）产品测试

独立的产品测试机构/小组是项目经理的最好助力。产品-测试小组是顾客的代理人，专门寻找缺陷。

（4）制定进度表

- 关键节点必须具体、独特、可度量。
- 重视每一次的落后，这是灾难的关键组成。
- 使用PERT图（与甘特图相比具有任务的依赖关系）
- 大的公司需要专门小组负责关注PERT图的更新、修改和报告。

（5）老板与项目经理的定位

- 减少角色冲突，老板必须区别行动信息和状态信息，需要规范自己，避免干涉项目经理在能力范围内处理问题的行动。这样有利于老板获得真实的评估结果。
- 问题棘手，项目经理难以应对，或者是项目经理没有权限调用相应的资源时，需要及时告知老板，避免问题无解。

8、软件中的根本和次要问题

（1）软件实体重要组成部分

- 数据集合、数据条目之间的关系、算法、功能调用等。

（2）软件的内在特性：复杂度、一致性、可变性和不可见性

- 复杂度
  软件实体可能比任何由人类创造的其他实体要复杂，没有相同的部分，有的话就合成子函数了。其他实体往往存在大量重复的部分。软件的复杂度让团队间的沟通非常困难，会导致产品瑕疵、成本超支和进度延迟。

- 一致性
  软件的开发目标就是兼容性。很多复杂性来自与其他接口的一致，对软件的任何再设计，都无法简化这些复杂特性。
- 可变性
  软件实体经常会遭到持续的变更压力。软件实体对应的是功能，而功能需求恰恰是最容易改动的，像业务逻辑会随着应用场景的改动和完善而变动，从而影响功能需求，进一步影响软件实体。
  硬件的更新或者新技术的引进也会影响软件实体。
- 不可见性
  软件具有无法可视化的特性。

9、书名的含义

- 人力（人）和时间（月）之间的平衡远不是线性关系，使用人月作为生产率的衡量标准实际是一个神话。

二、个人感悟

1、制度是根本

（1）好的制度可以保证大型项目的顺利进行，造就航空母舰级的公司；糟糕的制度确是寸草不生。 （2）**制度本质是对抗人性的弱点，充分发挥人的优势，最大化的减少群体间的内耗。** 好的制度比糟糕的制度在对抗人性的弱点，发挥人的优势，减少群体间的内耗方面表现的更好。比如好的公司制度，像阿里，可以有效打压腐败。类比国家制度，东方特色的集权制度比欧洲的民主制度能明显减少内耗，比如新加披的集权制度比法国的民主制度内耗小的多得多。在控制人性的弱点方面，中国的制度能强权控制毒品的泛滥，和美国比有明显优势。 （3）**制度影响收益和利益分配，利益分配又影响收益。** 华为的制度让公司利益往员工身上倾斜，这样保证招到优秀员工，同时提高员工积极性，和IBM化的项目管理制度等一同促进华为成功。像国内的360，利益分配对高层管理层都不公平，这也是18年，其管理层动荡的原因。像产学研落地的的科研中心、科研基地，本身制度将利益极大的倾斜到科研带头人身上，如果带头人过度摄取利益，该制度下，只能造就各种僵化的小公司。

2、公司优劣的评价

（1）老板和项目经理的职权划分要约束，越俎代庖天然制约公司的发展。 （2）程序员的工作是天然的重脑力活动，工作环境不能保证安静，效率必然低下。

3、项目成功的保证

（1）专业 相关的人员要有技术背景 （2）严谨 项目的需求定义，进度安排，进展把控等都需要严谨。

资料

1、《人月神话》精简PDF