第一部分 管理你的项目：

第一章：交付高质量的产品

即便是质量低劣的软件，其质量要求也要高于其他类型的人类书面文字。编程是一项严谨的工作，所以从事此项工作的专业人员都是在完成对质量要求极为苛刻的任务。唯一的问题是，根据历史发展趋势，未来的软件将远比今天的软件更大、更复杂，这就意味着即使保持目前的缺陷等级，我们在未来的工作也必须达到更高的质量要求。

在软件开发过程中，为了保证质量始终如一，必须遵循以下8个步骤：

1. 确立质量控制的策略、目标和计划；
2. 正确训练、指导和支持开发人员及其团队；
3. 确立和维护软件需求的质量管理过程；
4. 确立和维护软件工程过程的统计控制；
5. 审查、检查并评估所有的产品制品；
6. 评估所有缺陷，加以更正并用以识别、纠正和预防其他类似问题；
7. 确立和维护配置管理和变更控制系统；
8. 持续改进开发过程。

软件产品的质量应当被定义为产品对用户的有用性。

必须要记住，只有得到了清晰的需求，才可能开发出高质量的程序。

软件工程师的工作就是在计划成本和计划进度内交付高质量的产品。软件产品必须满足用户的功能需求，同时也要能可靠、始终如一地完成用户的工作。能完成用户的工作是关键。尽管对于程序的使用者来说软件功能非常重要，但是只有在软件能够运行时，这些功能才可以使用。

在你实现程序的其他任何目标之前，软件自身的大部分缺陷必须是你首先要解决的问题。

你首先要做的是了解这类在开发过程中引入的缺陷，并且尽最大努力防止它们出现。要想做到这一点，你必须熟练运用所使用的编程语言，全面了解你的开发支持系统，并且精通将要开发的这类应用程序。

缺陷是指程序中的错误，例如语法错误、拼写错误、标点符号错误，也就是不正确的程序语句。缺陷是一种客观存在的事物，是可以识别、描述和统计的。

对每一位软件工程师而言，缺陷都是应当高度关注的，不仅因为它们会影响软件产品的使用，而且一般软件公司超过一半的精力都用于发现和修复这些缺陷。由于用于测试的时间非常昂贵，并且很难事先预计，所以缺陷通常是导致项目成本和时间进度出现问题的最主要原因。

当软件程序得到了广泛的应用，特别是以一些设计者事先没有预想到的方式应用时，那些表面看起来十分细微的错误就可能会产生无法预知的后果。特别是对广泛应用的软件系统进行扩展以满足新的需求时，那些潜伏的问题就有可能暴露出来，一个看似微小的缺陷都有可能会引起灾难性的后果。

人们常常把软件质量看做是最终的结果或终点。事实并非如此，这是一段永远都没有终点的旅程。

目标提供了一个任务和焦点，帮助我们确定优先次序并忽略那些不重要的细节。为了完成重要的工作，首先要清晰地知道什么是你要努力去实现的。方向模糊，目标不明确，就是在浪费时间。

目标之所以重要，主要是基于以下两条原因：它们提供了努力的焦点，而且建立了一种优先次序。

要坚持让你的问题得到解决。很多情况下你会发现，其他人遇到的问题和你的几乎完全一样，只是他们羞于开口询问罢了。通过想你的管理者请教，明白无误地了解他们想让你达到的目标是什么，只有这样你才能真正履行好自己的职责。如果他们不能做到这一点，那么在开始工作之前，把你对目标的理解整理出来，然后与他们核对。之后要确信你和他们达成了共识。

第二章：为高质量项目制订计划

计划分为两种类型。第一种的基于时间段的计划，这个时间段可以是日历上的任何一个片段。阶段计划关心在这一段时间内你准备如何利用时间。第二种是基于行动的计划，比如开发一个程序或撰写一份报告。

在任何行业，工作稳定的关键是单位的财务健康，而财务健康的关键在于精确的阶段计划和产品计划。

如果你想成为一位有效率的软件工程师，你就需要知道如何制订计划。制订计划的关键是实践，因此要想得到最佳的实践，从现在就应开始制订计划，并且在你今后所有的项目实践中都要制订计划。

制订产品计划的第一步是清楚地定义要生产的产品是什么。

作为开发人员，我们要为自己的工作负责。在一个不现实的进度下，很难甚至不可能制订一个可行的计划。

由于计划不当而造成的种种问题会使所有人都蒙受损失：客户得到的是产品延期和开销加大，管理者不得不耗费更多资源，而开发人员则名誉扫地。为了成功，公司必须实现他们的承诺。而我们为了履行职责，必须制订精确的计划。

最终得出的计划必须满足特定的要求。就一份计划而言，有五条基本的要求：易于理解、清晰明白、详细具体、精确缜密、准确无误。

与其计划好未来所有工作的细节，不如只粗略划分后期任务，只有在需要时再对其进行完善。

第二部分 管理你的团队：

第3章：高效团队的基本要素

如果团队的进度计划不符合实际情况或者目标盲目乐观，那么在疯狂努力以达到目标的过程中，他们的行为常常可能失去理性。这种现象通常会对项目造成危害，并有可能导致其彻底失败。

尽管参与度在一定范围内波动是正常的，但是所有成员都要为团队的目标而努力，这一点十分重要。如果团队中有个别人明显没有真正去努力的话，整体士气就会受到影响。没有什么比在一个群体中有人公然逃避责任更具破坏性的了。李·艾柯卡把这种现象称为均等牺牲。

如果团队的进度计划不符合实际情况或者目标盲目乐观，那么在疯狂努力以达到目标的过程中，他们的行为常常可能失去理性。这种现象通常会对项目造成危害，并有可能导致其彻底失败。

凝聚力是指把团队成员紧密联结为一个统一的工作群体，无论在身体上还是在精神上都达到一致。一个有着高度凝聚力的团队，其成员之间的交流是自由但频繁的。虽然他们不必是好朋友，但他们配合默契、彼此尊重、相互支持。

人们通常认为好的交流者是那些表达清晰、辩才无碍并且能吸引注意力的人。这其实是一种误解。最杰出的交流者实际上是优秀的倾听者。在倾听并且真正理解之后，他们才能以一种直接解决你的问题和焦虑的方式进行交流。这种类型的交流总是以倾听为起点。

尽管在动荡阶段团队的反应通常比较情绪化，但他们的担心是真实存在的。项目描述常常是极不充分的，并且时间进度要求极其紧迫，根本不可能完成。在动荡阶段，团队努力渡过这个痛苦的过程，最终发现必须要理智地解决这些问题。一旦开始把精力集中到如何解决问题时，他们就会迅速建立起一种有效的方式来共同工作并完成任务。

在工作型群体中，团队成员主动进行工作。他们不会不择手段地争取职位或者操心谁做得最多。他们不会抱怨不合理的管理，也不会盲目服从团队领导者和辅导员。当他们认真思考该如何完成任务时，他们会具备非凡的创造力。

当一支团队的行为像一个过程型群体或对抗型群体时，它是无法有效进行工作的。这时团队成员把绝大部分精力放在了处理察觉到的内部或外部问题上。虽然团队看起来似乎是忙碌且精力集中的，但其成员却惶惶不安，他们知道团队没有在有效运行。

以下就是自主指导型团队的典型行为：团队成员会主动发现需要做什么并及时去做，而不用人告诉要做什么，为了能完成工作任务他们会去做任何需要做的事。对于工作，他们勇于承担，并决心把它做好。

事实上，无论在什么专业领域，专家和业余爱好者之间的区别就是有没有纪律。真是原意去排演、实践和不断提高，专家才成为专家。研究表明，世界级的艺术家与那些半途而废的人之间最主要的区别就在于其遵守纪律的行为。

第4章：做一位高效的团队成员

作为项目团队的一分子，你应当作出负责任的承诺，并且努力去实现你的承诺。这也是团队能运转的唯一方式，团队成员必须相信其他人能说到做到。为了履行在任务分配时作出的按时完成工作的承诺，你就要负责任地制订计划。

技术领域有一条不成文的规律，那就是所有的意外都将牵出更多工作。工程师根据经验在计划中已经考虑到一些意外，但他们的计划永远不可能完全精确。

目标对个人来说十分有用。有一点几乎毋庸置疑：没有目标就不可能去努力。没有目标，所有的努力似乎都是没有意义的，是在浪费时间。毕竟，如果努力不能让你达到任何目的，为什么还要费力呢？

项目团队提供的不仅仅是组织架构，它们还提供了一系列技巧和视角。团队结合了众多人的知识和经验，如果能被恰当地利用，这可以是一种强有力的资源。

所有的新思想都始自仅有一人的少数派。如果你就是这个人，那么你要对这个新思想和项目团队负责。很多情况下，只要你转而要求团队成员解释他们的逻辑，一些全新的观点就可能涌现出来，项目团队可能因此会走向一个完全出乎意料的不同方向。这种现象被称为增效。

你也许不赞同持异议者的立场，但是你应当确保他的立场已经得到了仔细研究。你也许是这个人坚持下去所需的关键盟友，虽然你的支持可能还无法改变决定，但重要的是要让团队详尽考虑和理解所有成员的观点。

自力更生在某些时候很重要，但是在危急关头依然试图不借外力却可能很危险。当遇到麻烦时，你应当请求并接受帮助。而当有人提出原意提供帮助时，应该认真考虑他的建议。很可能正是你自认为不需要帮助的时候，恰恰却是需要帮助之时。

没有人能把全部时间都用于设计、编程或测试等工作。如果工程师试图那样做的话，他们会失去工作的激情而且很有可能会犯错。角色职责带来的各种工作事实上提供了一些作为开发工作调剂的新花样。

一个积极进取的目标能起到激励作用，但并不是任何目标都可以如此。它一定是团队成员一致认为重要，也是他们感到能够实现的。

原则式谈判之所以有效的原因是，它避免了立场的两极分化。当谈判双方持相反的立场时，他们绝不可能达成一致，一方胜利就意味着另一方失败。从这种两难境地解脱出来的唯一方法就是将关注重点从立场转移到利益上去，询问双方追求的究竟是什么、他们真正想要的是什么。

由于每位成员都有帮助维护秩序的义务，所以你需要小心区分破坏性行为和真正的焦虑。如果有人受到了压制，他所关心的问题没有得到研究，那么应当为他提供支持。但如果有人是在故意破坏，那就要想办法支持团队领导者。

经验证明，团队中如果有一个人不务正业的话，就会影响其他所有人的表现。事实上，剔除不履行职责的团队成员通常会提升团队的整体表现。

作为一名团队成员，你要记住：你并不孤独。如果你坚持要做孤胆英雄的话，那么迟早有一天你会陷入无法自拔的困境。在迟迟不寻求帮助的过程中，你有可能已经破坏了团队实现目标的能力。

对软件开发来说，团队给予其成员的支持是决定性的。出来正式的团队角色外，要记住你还是团队的一名公民，有责任抽出一点时间去帮助他人。这就意味着你应当参与检查、提出建议、参与设计。几乎所有人都会偶尔从建议或帮助中受益，并且同行评审是产品成熟过程中绝对必需的一个部分。当你得到团队的支持后，你才更可能到达最佳状态。

提供有效支持的关键是，帮助你的团队伙伴相信他们自己的能力。有了自信，他们就有可能去努力完成优质的工作。他们也许会怀疑自己处理一些问题的能力，也许还需要帮助和指导，但要记住的关键一点是，人们能做到的远比他们自己认为的要多得多。

第5章：领导和指导你的团队

领导力决定成败。无一例外地，真正的创新性工作只有由杰出的领导者带领团队才能完成。不过有趣的是，那些领导者在先前通常还只是十分普通的开发人员，就像你我一样，但当他们担任领导职务之后，去的的成就却令人瞩目。

刚刚加入团队，没有经验的开发人员常常不愿意去做他们不熟悉的工作，他们缺乏承担风险的自信和工作保障。而有经验的开发人员通常面对新的技术挑战时会感到兴奋，他们渴望学习到新的方法和技能。不过另一方面，有经验的开发人员很有可能会对重复做相似的工作缺乏兴趣和积极性。

当管理者诉诸威胁或恐惧时，实际上是在降低部属的马斯洛需求层次。此时团队成员关心的是成员资格甚至是个人安全，而不是被他人尊重和自我实现了。

即使没有迫在眉睫的软件创建和发布，你也能制造出一种紧迫感。不过，为了激发积极性，这种紧迫感必须是真实、可信和公开承诺的。

应该这样看待过程：你所想要的是对工作有用的一个过程，而不是由别人定义的标准化的框架。

所以应当等等那些安静的成员。很多情况下他们默不作声，是因为他们正在倾听或思考。当给他们机会时，他们通常会有很多话要说。他们的发言会十分中肯，而且大多数情况下他们的发言内容都是重要和有用的。

使用数据的一个重要原因是，它能够帮助项目团队作出正确的团队承诺，并且还能帮助他们说服管理者该计划是正确和积极的。此外，关注事实和数据会增强团队成员的参与程度，它能创造一个人人机会均等的局面。

在动荡阶段，辅导工作的关键是让团队成员发泄他们的怒气和愤闷，然后着手解决他们的焦虑。不要让项目团队匆匆渡过这一阶段，而是要帮助他们理解和处理他们感到沮丧的事项。一旦看到无法回避，并且问题也不会自动解决的时候，他们就会同意制订一个计划。而一旦制订了计划，他们就会想要马上开始执行。

当管理团队以一种公开、坦诚的方式在一起工作时，其结果一定是最佳的。他们通常是组织中最有能力的人，同时也是掌握信息最多的人。作为一个群体，他们最能平衡技术和业务上的争议，并且能制定出行动的正确计划。当他们理解了决策，相信它们是正确的，并且能向他们的队员解释决策时，这些专业人士就知道他们的需求已经被考虑到了，因而极有可能会充满活力地投入到决策执行中去。

第三部分 管理你的领导

1. 讨论项目并捍卫你的计划

如果有人能证明你的估算有偏差，那么要及时进行调整。只要他们有真实的历史数据作为支持，只要那些数据与你的项目相关，就应当仔细考虑新的事实。不过无论如何，不要急于现场作任何决定。

如果你认为日程进度将会比管理者预期的要长，不要在此时就与他发生争辩。你还没有充分的事实和论据来帮助你赢得这场争论，这只会向管理者表明你没有一个端正的工作态度。

如果没有同类经验、没有专业指导、没有引进计划、没有得到管理者的认可，就盲目地采用一些诱人却自己知之甚少的新方法，这不可避免地都会成为一个错误。你优先级最高的事就是当前的工作任务。在作任何改变之前，一定要弄清楚这种改变可能会对工作产生什么样的影响。

你甚至应该在事先准备好几份备选计划。此时的关键是，要反应迅速，对工作的考虑要细致周到，并且在作出任何承诺之前都要制订一份计划。要仔细检查所有备选方案，并竭尽全力去实现管理者的目标，但是要抵制一切没有计划支持的进度或资源削减。

如果支持不足的话，工程师就只能花费一小半的时间和精力用于完成工作任务，而完成任务才是培训和雇用他们的目的。之所以出现这种现象，主要原因就是没有足够的支持，工程师不得不自己支持自己。更重要的是，极少会有工程师乐于或者擅长兼职去做这类办公室的杂务工作。

因此，要留心组织内的指挥链并从你的直接领导者着手。不要做任何你的管理者不知道或者不同意的事。当然这并不是说，在你开始工作之前，管理者必须要知道你的所有步骤，而是说你的工作要对他们完全公开并且绝对诚实。向管理者解释你的方法，确信你们双方都理解这个计划，并且双方对于你可以自主决策的权限取得了一致。

真正的考验是要看在做出决策时管理者是不是真的考虑了你的需要和你的意见，仅仅假装听取你的看法并不能改变独断专行的事实。典型的独断专行者事实上是在说：“让我们都妥协一下，按我说的去做。”

津巴多由此得出结论：只要给一个普通人一份卑微的工作，而这份工作又有一点极小却绝对的权力时，他通常就会以独裁主义者的方式行事。

与头痛相比，在软件过程中，绝大多数组织的问题可能更像是中风或脑瘤。虽然你可能不得不把改进说成是解决短期问题的方法，但是一旦你引起了高层某些人的关注时就应当尽快转向战略问题。

在估算这些成本的过程中，要记住一条关键的指导原则：别人评判你的陈述，是看其中的最薄弱环节。

如果你是很小心地挑选改进措施，并在每一步实施过程中都寻求到支持，那么随着时间推移，你很可能得以不断推进改进方案。接下来，你就可以逐渐把短期战术举动转变为长期战略性努力。

在很多方面，管理者在意的东西与团队的基本利益是一致的。他们想得到技术熟练和感到满足的雇员，而且他们需要稳定和可靠的员工。不过，在构建具有凝聚力和提供满足的团队环境方面，双方的利益则有差异。虽然几乎没有管理者会反对这样一种环境，但通常情况下他们不会在这上面投入过多思考或赋予它很高的优先级。

第四部分 管理你自己

1. 控制你的工作

有一个明确定义和精确衡量的过程，你就不仅可以知道今天完成的工作情况如何，而且还会看到该如何提高，从而在软件开发期间保持持续不断的进步。

虽然软件开发的确是一个富有挑战性的行业，总是要面对紧迫的日程进度、不断更改的需求，但这些问题都是能够管理的。不过，只有在你知道如何管理它们时，那些问题才是可以管理的。

单纯的衡量不会产生改进，单纯的努力也不会。你的工作方式在很大程度上决定了你所得到的结果。如果依然沿用旧的工作方式，你无疑将得到和以前一样的结果。

过度的压力会产生破坏性的后果。它让人们产生焦虑，想象出一些不存在的问题和困难。压力导致对未知（且常常是虚幻的）问题过于焦虑，而不是帮助你以一种有序、建设性的方法去解决它。压力之下，如果你视假为真，你的举动可能会给项目、组织甚至自己的自尊造成不良的影响。

你在本周利用时间的方式很有可能与上周一样。一般情况下，你在上周使用时间的方式会与你在未来几周使用时间的方式近似。

让计划精确的关键就是不断制订计划，并把每一个计划的时间与随后的真实时间进行对比，这样你就能知道如何制订更好的计划。要想管理好时间，就是要计划你的时间并遵守这个计划。

当工程师认为一个项目将要陷入麻烦时，它几乎肯定会陷入麻烦。当人们保持沉默并且寄希望于他人会意识到问题时，问题就会逐渐变得更加难以处理。你每多等待一天，就多浪费了解决问题的一天时间。

常见的情况是，软件承诺几乎只是以良好的愿望为基础。即使双方都真正想履行承诺，但仅仅只有良好的愿望并不能为合理的承诺提供理智的基础。

信守承诺是一种精神状态。不管出于什么原因，你已经着手做某件事了，并且你认为应当去做它。不过承诺不仅仅是你想要做某件事，这件事也是别人想让你去完成的。

地位本身并没有什么错，但它们都是暂时的。也许在一段时间内你身居高位，但迟早你要从位置上退下来。失去地位会变成一场灾难，有些人在第一次失去重要职位后就变得萎靡不振。人们很容易把工作的重要性和个人的重要性混为一谈。

完成高质量的工作后，你会感到自豪。即使无人知晓，你也知道自己的工作是一流的，并会为自己进来全力而感到十分满足。令人惊奇的是，高质量的工作总是会为人所知。虽然可能需要很长一段时间才得到人们的承认，但这是迟早的事。不论你是否意识到，你都会因为你的工作质量而赢得他人的信任。

你的发展受限于你增强自身才干的能力。因此要在每一个项目中锻炼自己的才干，主动衡量和观察自己的工作。你不能停滞不前，所以应当把每一个项目视为锻炼才干的机会，而不是仅仅把你的才干视为完成项目的工具。

我们生命的终点都是一样的，所以需要把精力集中在这段旅程上。如同一个过程，一旦你想好要如何生活，其他的就纲举目张。执著追求卓越，你才有可能臻于卓越。这样才不枉在人世走这一遭。

第8章 学会领导

如果你对团队能否圆满完成任务有所怀疑，即使你什么也没有说，这种疑虑还是会暗地里影响你的行为。当团队成员感觉到你不信任他们时，他们几乎总是会失败。

作为领导者，你的行为人所共睹，并会被团队成员视为榜样。李·艾科卡曾经说过：“领导者的速度就是团队的速度。”你不可能期望团队成员比你更有责任心、工作更努力或更仔细。

领导力低下非常明显的症状是，高层管理者在思考问题时以自我为中心、眼界短视狭隘。这些管理者太过于关注他们当前的议题、问题和利益，以致看不到自己个人狭窄视野之外的天地。

领导力低下的另外一个常见症状是那种我称为“官僚惰性”的现象。在官僚主义的领导风格下，没有人会去试图消除工作中的障碍，或者根据当前任务的优先级、技术、目标去再次检查组织的资源分配状况。

团体的集体智慧要比那些哪怕是最富技巧、最有才能的个体更有成效。因此要利用好你的团队，它们需要领导，也想得到领导，并且会很乐于帮助你去做好领导。

我们中的绝大多数人都喜欢从事技术工作，我们四处寻找开发职位，是因为喜欢去做一些富有创造力和挑战性的事情。我们也喜欢看到辛勤努力后的成果，特别是当产品能以我们预料的方式运行时更是如此。不过，如果有人像对傻瓜一样对待我们，或者认为我们不会思考，我们就会失去动力和创新的火花。

领导者和管理者之间最主要的区别就是，管理者命令员工服从他们的指令，而领导者是带领他们完成任务。

你是领导者的唯一判断标准是：你正在领导而他们追随者你的领导。如果你因为突然想到了解决一个重要难题的办法而在半夜惊醒，不妨与你的团队分享你的激动和热情。当你的员工有了一个绝好的主意或完成了一件关键的任务时，不要吝啬你的鼓励，和他们一起庆祝。你要成为团队的一部分，并且让团队成为富有回报和令人愉悦的工作场所。

当员工不是内在地被工作所激励时，他们的创造力会用在千方百计以更少的工作获得更多的报酬上。由于这种工作态度几乎无助于项目的成功和创造性，所以你的目标必须是成为变革型领导。

虽然成为一位领导者不会让你变得更聪明，也不会让你拥有任何特别的洞察力，或增加了你的知识，但现在你承担了一份责任，从此之后，这种责任要渗透到你的一言一行中去。

作为一名领导者，你的确有一些有限的权力，而你必须决定在哪些地方使用它。尽管你看到的很多问题的确让人厌烦，但要把精力集中在那些真正会影响工作的事项上，而不是抱怨其他人。