

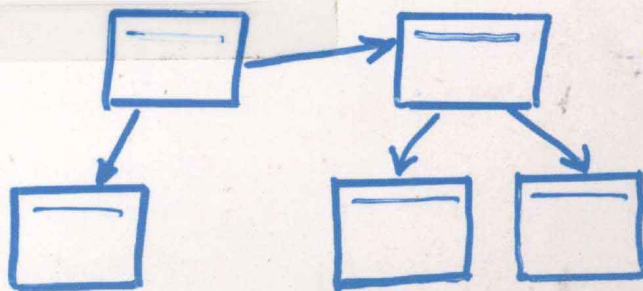
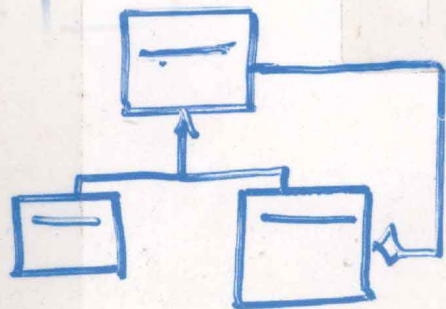
管好团队做对事

——软件企业成长手册

GROWING SOFTWARE

PROVEN STRATEGIES FOR MANAGING
SOFTWARE ENGINEERS

[美] Louis Testa 著
郭稚晖 译



人民邮电出版社
POSTS & TELECOM PRESS



管好团队做对事 ——软件企业成长手册

GROWING SOFTWARE

PROVEN STRATEGIES FOR MANAGING SOFTWARE ENGINEERS

[美] Louis Testa 著
郭稚晖 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

管好团队做对事：软件企业成长手册 / (美) 特斯塔 (Testa, L.) 著；郭稚晖译. — 北京：人民邮电出版社, 2011. 5
ISBN 978-7-115-23127-7

I. ①管… II. ①特… ②郭… III. ①软件—电子计算机工业—工业企业管理 IV. ①F407.67

中国版本图书馆CIP数据核字(2011)第005978号

版权声明

Copyright © 2009 by Louis Testa. Title of English-language original: Growing Software, ISBN 978-1-59237-183-1, published by No Starch Press. Simplified Chinese-language edition copyright © 2010 by Posts and Telecom Press. All rights reserved.

本书中文简体字版由美国 No Starch Press 授权人民邮电出版社出版。未经出版者书面许可，对本书任何部分不得以任何方式复制或抄袭。

版权所有，翻印必究。

管好团队做对事——软件企业成长手册

-
- ◆ 著 [美] Louis Testa
 - 译 郭稚晖
 - 责任编辑 傅道坤
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本：800×1000 1/16
印张：17.25
字数：371千字 2011年5月第1版
印数：1—4 000册 2011年5月河北第1次印刷
 - 著作权合同登记号 图字：01-2009-7286号
ISBN 978-7-115-23127-7
-

定价：39.00元

读者服务热线：(010)67132705 印装质量热线：(010)67129223

反盗版热线：(010)67171154

内 容 提 要

软件行业中最经典的语录就是“唯一不变的就是变化本身”。作为中小型软件公司的技术领导，在迅速应对行业、市场以及公司内部调整变化时，在做好队伍建设，产品定义，工程实施的同时，既要贯彻高层主管们所制定的战略指导方针，也要处理好各级部门与同事的关系。这些技能与素养的培养对于那些刚刚从工程师队伍成长起来的技术经理们来说是一个很大的挑战。

而本书就是站在这样一个特定的角度，教您如何培养这些技能与素质，以帮助您解决软件公司运营过程中出现的日常危机，并指导您如何打造并领导一个高效的团队，如何定义你们的产品，如何实现从客户到 CEO 之间各类人员的精诚合作，从而确保完成高质量的交付产品。本书包含了以下内容：与你的 CEO 和执行团队一起有效工作；提升开发团队效率并激发他们的工作热情；评估你们的软件方法用以提升开发效率并防止失误；利用产品原型来弥补市场与工程直接的空隙；排除技术定时炸弹的隐患等。

作为有着 20 余年项目管理经验的高级技术经理，Louis Testa 先生在本书中融入了大量的实际案例和指导意见，使得软件管理行业的新人可以在字里行间体会他的谆谆教诲，找到前行的路标。而具有一定管理经验的项目经理通过阅读此书，也可以更新其陈旧的管理经验，打破其原有的陈规旧俗，从而对软件项目管理的认识达到一个全新的高度。

致 谢

正是因为付出了大量的心血，才使得我的第一本书如此有趣。然而，无论我个人付出多大的努力，如果没有包括家人和朋友在内的众人给予的帮助与建议，本书将不值一读。

没有我妻子 Edie 以及 4 个孩子 Logan、Kevin、Kerry 和 Brady 的鼓励，我不可能写完本书。Edie 的忠告和建议促使我重写了许多章节并填补了其中的空白。

在此，要特别感谢 Clayton Greer，他对本书进行了细致的技术审核并提供了大量有价值的建议；感谢 Anita Maria Gutierrez 对本书全稿广泛而全面的审核，以及从编辑角度所提供的见解和改进建议；感谢 Jef Bell 对本书全面而透彻的审核与积极建议，并提供了让本书更有说服力的创意；感谢 Mike Portwood 对本书主题富有见解的建议以及为审核本书花费了大量时间。

此外，对 Bob Tidwell、Curt Frye、Paul Irvine、Gordon Huntsman、Miki Tokola、Rick Sanstrom 和 Dylan McNamee 提供的宝贵建议与信息致以诚挚的谢意。

最后要特别说明的是，与 No Starch Press 的团队人员一起工作的感觉很美妙。他们是 Megan Dunchark、Tyler Ortman、Lisa Theobald 和 Adam Wright。任何有志于图书出版的技术作家，都可以联系他们。我向你们隆重推荐（我想我做到了）。

前言

在许多成长中的小型公司中，项目经理们在执行 CEO 战略意图的同时，往往还得处在不得不与技术团队和其他高级经理打交道的位罝。通常情况下，即使当工作的非技术因素已经最大程度地影响公司的成败时，开发经理们依旧仅仅关注于技术。伴随着公司的成长，曾经看上去很小的问题也在不断扩散，最终引发成为主要灾难。我写本书的目的，就是为那些新上任的开发经理们提供建议，告诉他们如何成功地应对这些挑战。

小公司中的开发经理在很多方面都不同于在一个稳定的大公司中的同等角色。例如，小公司的开发经理通常不仅要和开发人员一起支撑亟待成熟的产品，还必须与被公司成长过程中面临的挑战所吸引的技术大牛共事。但最重要的是，小公司的开发经理必须对员工、产品、过程、规划、技术以及客户等诸多因素给予广泛的关注。

相对而言，大公司通常支持多种现有产品，他们的过程定义通常比较清楚而且相对稳定。大公司的政策通常限制了开发经理所用的工具、技术及实现方式。相对于小公司的同等角色，大公司的开发经理工作范围更明确而狭窄。

本书作为一本实践指南，为那些经历过初始生存阶段且正在试图进一步发展壮大的小公司中的开发经理们提供了实用的指导原则。其目的是帮助管理者预见问题并在其变得棘手前及时进行处理。这里描述的技能主要适用于销售产品级软件或提供软件服务的小公司，并不针对软件咨询业务。本书不仅提供了总体建议，也提供了具体解决方法以及详细的模板和电子表格，这些内容都非常有助于开发经理们将基本概念应用于具体实践。

由于涉及范围较广，本书的书写方式采用了指令式的风格（**prescriptive style**）而不是议论型 **argumentative** 风格——也就是说，不建议采用广泛的论据来说明这些良好的工作技能如何到位。这方面的资料将大大扩大本书的范围，但同时也会降低它的可读性。

为方便起见，笔者使用“开发经理”和“开发管理人员”贯穿全书，这些术语用于指代高层软件/工程管理人员——不论是首席技术官（CTO）、工程副总裁、工程总监，还是高级工程经理这样特定的工作职位。这个人既要从事软件工程师的管理，还要担负质量保证、文档说明以及项目管理等团队的管理。虽然本书的目标读者是开发总监，不过非技术类管理人员也同样会受益于本书所描述的问题和解决方案。

本书的章节结构及相关说明

为便于参考，本书分为如下主要部分。

- 开发团队。
- 产品和技术。

- 工程之外。
- 为项目、过程及质量管理制订工作流程。
- 规划未来。

虽然本书是按顺序逐个主题逐层深入的，但读者还是可以跳过一些章节直奔感兴趣的

主题。

公司发展阶段

公司从启动到发展成熟是分阶段成长的。本书将涉及其中的一个或多个阶段。表 1 是根据公司的规模和产品的成熟度对发展阶段所做的定义。

表 1 公司启动发展的阶段

阶 段	公司规模 (员工数量)	客 户
启动	少于 12	0~2 个, 没有主要客户
立足	12~40	3~5 个, 1 个主要客户
发展	40 以上	6 个以上客户, 2 个主要客户

现实生活的记述

本书提供了用于解释关键点的现实情况的叙述短文，这些短文散布在全书各处。虽然出于一致性的考虑，这些故事是以第一人称来写的，但它们是个人及其他人的经验汇总。正文中隐去了公司和个人的名字。

电子表格

电子表格示例用来说明解决特定类型问题所需的信息汇总、分析及展示技术。每一个电子表格都在正文中予以说明和描述，它们可以从 <http://www.nostarch.com/growingsoftware.htm> 下载，并用在 Microsoft Excel 或 OpenOffice.org Calc 中。这些示例的主要目的是让读者利用简单的电子表格去分析和解决潜在的问题。

虽然这些电子表格可以用正常的方式使用，许多情况下还是需要你自定义电子表格的样式布局。你自己也可以通过输入所有的字段来重建电子表格。箭头所指单元格中可以输入计算公式，并可以在示例的右边或下边给予说明。在适当情况下，可以按照电子表格中所附注释的描述来复制整行或整列公式。图 1 中给出了示例和说明。

输入公式时，在将公式从一个单元格复制到另外一个单元格时，要注意美元符号（\$）会对公式读取数据的方式产生影响。这些影响不包括公式被复制后，美元符号导致的计算错误。

	A	B	C	D	E	F
1	周	1	2	3	4	5
2	总计 #	2	3	4	2	3
3	级别	3	5	4	3	3
4		2	2	3	3	4
5			1	3		2
6				3		
7	当天日期	6-Jun				
8	成本总计	\$143				

=SUM(K12:K19)
注释：复制这一行

=COUNTA(F3:F6)
复制到 B2:E2

图 1 本示例列举了单元格的操作说明

对于电子表格示例来说，正确的格式同样很重要。默认的输入格式是“常规”格式，这种格式并不会以最适当的方式显示所有数值。上例中隐含了一些格式：日期应该采用日期格式；货币应采用小数点位置带零的货币格式；数字应采用的格式是根据所需要的小数位数进行四舍五入。

Excel 与 OpenOffice.org Calc 的显著差异

- OpenOffice.org Calc 使用分号 (;) 分隔公式中的字段，微软 Excel 使用逗号 (,) 分隔。本书在所有的公式中采用逗号分隔所有的字段。如果使用 Calc，要使用分号进行替代。例如，Excel 公式 =WORKDAY(B3, C3) 在 Calc 中则为 =WORKDAY(B3; C3)。
- 第二个显著差异是使用表间引用来引用其他工作表的单元格。在 Excel 中使用感叹号 (!) 实现表间引用，而在 Calc 中使用的是句点。例如，在 Excel 中引入 Eng 表格的公式是 =Eng!H3，而在 Calc 中则是 =Eng.H3。

模板

本书也提供了由虚线框围着的示例模板，以便于读者复制用做初始模板。读者在填充模板时，可随时将模板的说明予以删除。

目 录

第 1 部分 开 发 团 队

第 1 章 入门	3	2.7 教导问题员工	25
1.1 在新工作中找到你的出路	4	2.8 考核与评价	26
1.1.1 立即着手处理	4	2.8.1 创建考评	26
1.1.2 经历初始培训	5	2.8.2 提交考评	26
1.1.3 收集信息	6	2.8.3 提交滞后以及缺陷考评的 不利影响	27
1.2 了解人	7	2.9 附加读物	28
1.3 不愿透露信息	8	第 3 章 创建一个高效的开发团队	29
1.4 认同企业文化	9	3.1 有效的团队组织	30
1.5 学习技术、过程和产品	9	3.2 程序员的效率	32
1.6 了解客户	11	3.3 办公空间	34
1.7 了解公司的业务流程	11	3.4 如何让其他团队与工程队伍沟通 顺畅	35
1.8 回归重点	12	3.5 新经理, 旧习惯	36
第 2 章 管理开发团队	13	3.6 富有乐趣	36
2.1 理解你的核心价值	13	3.7 附加读物	37
2.1.1 信任	14	第 4 章 扩充软件团队	39
2.1.2 灵活性	15	4.1 设计一个筛选过程	40
2.1.3 诚挚	16	4.2 面试特长	40
2.1.4 机密性	17	4.2.1 技术能力	41
2.1.5 尊重	17	4.2.2 业绩	41
2.1.6 授权	17	4.2.3 文化融合	41
2.2 与你的团队沟通	18	4.2.4 工作习惯和喜好	42
2.2.1 一对一	18	4.2.5 行业经验	42
2.2.2 项目沟通	19	4.2.6 个人能力	43
2.2.3 团队会议	19	4.2.7 沟通能力	43
2.3 解决冲突	20	4.2.8 个性	43
2.4 培训	21	4.2.9 热情	43
2.5 指导	23		
2.6 激励你的团队成员	23		

4.2.10 解决问题的能力	43	4.3.5 马拉松式面试或压力	
4.2.11 幽默感	44	面试	46
4.3 汇总	44	4.3.6 打了就跑的面试	47
4.3.1 电话面试	44	4.3.7 午餐委员会面试	47
4.3.2 办公室面试	45	4.3.8 评审会	47
4.3.3 指导你的面试小组	45	4.3.9 做出选择	48
4.3.4 面试中应避免的话题	46	4.4 附加读物	49

第 2 部分 产品和技术

第 5 章 定义产品	53	6.2.3 发布后评审	77
5.1 产品定义过程	54	6.3 发布版本的标识	78
5.2 产品定义内容	56	6.3.1 版本发布三级编号	
5.3 整体产品概念	58	标识	79
5.4 利用原型定义产品	59	6.3.2 补丁发布	80
5.4.1 总体原型	59	6.3.3 构件编号与版本编号的	
5.4.2 快速和灵活的方法	59	对比	80
5.4.3 可点击的用户界面原型	60	6.3.4 跨发布版本的编号	81
5.4.4 用户界面设计	60	6.3.5 乡绅的软件	82
5.5 与市场部门建立联系	61	6.4 附加读物	83
5.5.1 避免不良的关系	61	第 7 章 评估你们的工具和方法	85
5.5.2 保持市场和工程团队在		7.1 备份知识产权	86
一起	62	7.1.1 备份频率	87
5.5.3 建立平衡关系	62	7.1.2 异地备份	88
5.6 客户对产品的认识	63	7.1.3 仅在磁盘上备份	89
5.7 在 α 版本发布中改善产品	65	7.2 创建和管理开发文档	89
5.8 了解现有产品的组成部分	66	7.3 源代码版本控制	91
5.9 附加读物	67	7.4 软件构建方法与时机	93
第 6 章 驱动版本发布	69	7.5 软件发布过程	94
6.1 版本发布计划	70	7.6 缺陷跟踪系统	95
6.1.1 版本发布时间表	70	7.7 选择合适的开发工具	96
6.1.2 早期版本发布策略	71	7.8 附加读物	98
6.1.3 计划产品淘汰	73	第 8 章 评估你们的技术	99
6.2 版本发布过程	75	8.1 系统文档	100
6.2.1 版本发布标准	75	8.2 系统可扩展性	101
6.2.2 过程步骤	76	8.3 故障模式	102

8.4 错误处理和消息	103	8.8 安全	107
8.5 系统的灵活性与可维护性	104	8.9 数据报表与分析	107
8.6 整合入系统的第三方 软件包	105	8.10 国际化支持	109
8.7 系统应用程序接口	106	8.11 着眼重点	109
		8.12 附加读物	109

第3部分 工程之外

第9章 与你的公司一起工作	113	10.1.4 影响决策	124
9.1 企业文化和做法	114	10.2 与执行团队合作	124
9.1.1 企业风格	115	10.2.1 解决冲突	125
9.1.2 管理风格	115	10.2.2 用信赖打开你们的隔阂	125
9.1.3 会议风格	115	10.2.3 敞开沟通	126
9.2 处理团队内部问题	118	10.2.4 有效合作	127
9.3 增进同僚关系	118	第11章 倾听客户的声音	129
9.4 尊重工程团队	119	11.1 客户满意	130
9.5 附加读物	120	11.2 客户会议	131
第10章 和CEO及执行团队一起工作	121	11.3 搞定交易	132
10.1 支持你的老板	122	11.3.1 销售团队什么时候会过度 承诺	132
10.1.1 确保目标明确	122	11.3.2 寻求报价	133
10.1.2 提供有用的信息	123	11.4 支撑的要求与客户的需求	134
10.1.3 沟通	123		

第4部分 为项目、过程以及质量制定工作流程

第12章 项目评估	139	第13章 启动一个项目	149
12.1 建立一个评估	140	13.1 理解目标	150
12.1.1 创建任务列表	140	13.2 集结项目团队	150
12.1.2 建立任务评估	141	13.2.1 换人	151
12.1.3 考虑评估偏见	142	13.2.2 游戏延迟	152
12.1.4 创建评估	142	13.3 设置优先级	152
12.1.5 编写和交付评估	144	13.4 选择一个框架	153
12.2 采集原始项目数据	145	13.5 制定时间表	153
12.2.1 利用电子表格汇总数据	146	13.6 创建一个项目计划	154
12.2.2 工程维护和经常开支	147	13.7 启动会议	155
12.3 附加读物	148	13.8 附加读物	155

第 14 章 项目执行与跟踪	157	16.2 分析流程模型	188
14.1 一个项目的执行管理	157	16.2.1 在操作中进行流程分析	188
14.2 项目跟踪方式	159	16.2.2 利用模型来改进流程	191
14.2.1 甘特图	160	16.2.3 和其他团队一起工作	193
14.2.2 项目跟踪电子表格	163	16.3 坚持不懈地走下去	193
14.3 变更控制流程	166	16.4 附加读物	194
14.4 风险管理	168	第 17 章 理解质量保证	195
14.5 附加读物	170	17.1 质量的重要性	195
第 15 章 设计一个软件开发过程	171	17.2 质量定义	196
15.1 软件开发过程中都涉及哪些 内容	172	17.3 注重质量	197
15.2 开发过程的类型	172	17.4 质量评估	198
15.2.1 临时性过程	173	17.4.1 质量保证团队	199
15.2.2 瀑布过程	173	17.4.2 QA 的工具和环境	200
15.2.3 瀑布过程改进	174	17.4.3 QA 活动	204
15.2.4 迭代过程	175	17.4.4 QA 过程	207
15.2.5 螺旋过程	175	17.4.5 缺陷的等级评定过程	207
15.2.6 敏捷过程	176	17.4.6 创建相关功能及用途并 予以说明	208
15.2.7 其他过程和方式	179	17.5 QA 指标	211
15.3 自定义一个过程	179	17.5.1 版本发布后每周所发现的 缺陷统计	211
15.4 选择一个过程	180	17.5.2 每周缺陷权重统计数	212
15.5 引进一个过程	180	17.5.3 客户发现缺陷的权重	214
15.6 附加读物	182	17.5.4 测试运行通过百分比	214
第 16 章 流程改进	185	17.5.5 测试通过中所发现的 缺陷	215
16.1 建立一个流程模型	186	17.5.6 测试期间每周所修复的 缺陷	215
16.1.1 定义流程边界	186	17.6 质量与生产力方面的缺陷 影响	215
16.1.2 列出流程步骤	186	17.7 附加读物	217
16.1.3 创建一张流程图	187		
16.1.4 估算时间范围	187		
16.1.5 建立电子表格模型	187		
16.1.6 验证模型	188		

第 5 部分 规划未来

第 18 章 确定发展方向	221	18.2 创建整体产品	223
18.1 听取市场部门的意见	222	18.3 化解技术上的定时炸弹	223

18.4 筹划技术检修	225	7 结论	249
18.5 优化客户安装程序	226	附录 B 国际化	251
第 19 章 发展战略及路线图	229	1 需要考虑的国际化问题	252
19.1 建立产品路线图	230	1.1 翻译人员的编制和成本	252
19.2 对选择进行评价	231	1.2 数据库要考虑的问题	252
19.2.1 成本-效益核算	232	1.3 国家和语言需求	253
19.2.2 成本-效益对比	233	1.4 货币流通问题	253
19.3 创建单页纸的评估	233	1.5 日期、公制度量标准以及 尺寸问题	253
19.4 附加读物	235	2 国际化的最佳实现方式	254
第 20 章 继续前进	237	2.1 区域设置	254
附录 A 软件公司的组织架构	239	2.2 翻译过程	254
1 公司任务	240	2.3 质量保证	254
2 典型的个体公司	240	2.4 数据库以及导入/导出	254
3 典型的两人公司	241	2.5 翻译公司	255
4 12 人的软件公司	241	2.6 用户接口	255
5 24~50 人的软件公司	243	3 小结	256
6 100 多人的软件公司	243	附录 C 企业工作流程示意图	257
6.1 层次管理结构	245	1 创建一张简单的工作流示意图	258
6.2 矩阵型组织	246	2 工作流实例	258
6.3 小型产品团队	247		
6.4 灵活的项目团队	248		

第 1 部分

开发团队

开发经理们往往只关注公司的技术。虽然技术非常有趣，但优秀的经理人必须首先了解和他一起工作的同事。只着眼于技术而忽略人的作用，就如同在训练一个棒球队时将所有的时间花费在测试最新、最大的球上一样。

本部分包含了启动新的工作，和开发组织一起工作，以及成功扩展你的团队。不要担心，我们也会谈论技术，后面会讲到的。

第 1 章

人 门

想象一下你沿着一些新的路径在树林中远足旅行。天气很好，你也很放松。随着不断前行，你发现景色开始变化，你进入了未经开发、崎岖不平的荒野。最终，你停下来坐在一块大石头上并吃了点零食。

当你在吃的时候，你意识到你不知道自己身处何处。你忘了带一张地图，也忘了规划路线和做上路标。现在你突然感到了紧张和恐惧。你迷失了方向。你想走得更快些——无论是按既定方向前行还是转身回去。但你知道，迷失在树林中的时候，惊慌且任意方向上奔跑只能犯更大的错误。

你记得一些荒野求生的建议：当你迷失的时候，先停下来评估方位，然后行动。你已经停止了前行。你要评估自己的处境并问自己几个问题：我怎么知道我在哪里？我的背包中是否带有工具、地图或其他有用的装备？从出发到现在已过了多长时间？预计多长时间能平安返回？下一步，你制订一个计划——然后采取行动。

现在，换一个场景：你在一个在新兴领域迅速成长的小公司中担任工程副总裁。虽然你从公司创建伊始就伴随其成长，但它已不再处于你所熟知的起步阶段了。你感觉像没带地图迷失在树林里。就像徒步旅行者，任何迅速而匆忙的行动都将使你的处境更糟，所以不要恐

慌，先停下来，评估局势，尽可能汇总信息，制订策略，然后行动。

另一方面，如果你在一个成长中的公司开始一份新工作，你也许感到你陷入了树林之中——失去了最初跋涉的欢乐。同样的方法适用于这种处境，不管怎样，在你采取下一步行动前先停下来纵览全局并收集你能掌握的信息。不论你是第一次迷失还是再次迷失，采取如下步骤，你就可以找到出路。

1.1 在新工作中找到你的出路

伴随着技术型公司增长的波动性，忠于公司的高级职员会逐步减少。当一个公司业务重点发生转折或改变的时候，开发经理们通常开始找寻新的工作。当然，获得新工作的第一步，始于面试。

面试中，面试官将尽力描绘他们公司的美景，如同薄雾笼罩的莫奈名画。而一旦你开始工作，则图画开始扭曲，甚至有些像毕加索的作品。面试中所描述的微妙的小问题变成了你亟待解决的巨大危机。表 1-1 中以调侃的口吻比对了面试说词与现实之间的差异。

表 1-1 面试与现实的对比

面 试 说 词	现 实 情 况
我们有一些很小的质量问题	该产品是一场灾难。只要你上了贼船，就会为所有的错误承担一切指责
我们需要提升我们的交付承诺	没有产品可以如期交付，由于市场及其他高层不能下最后的决心，公司在整个开发过程中需要不断修改新产品功能。即使这样，所有的人都希望按期交付
我们有一支伟大的工程师团队，只是需要一些指导	团队成员正在大厅里尖叫着讨论，至少有一位工程师由于不够格要被开掉
我们是一个有紧迫感的公司，有干劲十足的员工	你那个“懒惰的”的开发团队正期待着混过工作午餐、每一个夜晚以及大多数周末——除非你干掉他们

在你加入公司走上令人生畏之路时，你需要充分利用现状。

1.1.1 立即着手处理

在你任开发经理的一开始的日子里，急迫的问题需要立即引起你的注意。一些问题自从上任经理离职之日就开始堆积在那了，在你桌上堆放着延期数月的决策方案。在沼泽中跋涉是对处理这些积压工作最形象的比喻。

在专注于这些问题时，你可能会感受到压力，这会危及你长期的成功。只关注急迫的问题，会让你失去从公司、产品、同事以及你的团队中学习成长的机会，所以要做一次深呼吸看一看更广阔的天地。

如果在解决悬而未决的问题与处理日常危机之间分割你的时间，你会最终减少一些重复出现的问题，这样会提升团队的效率。每天花一部分时间从公司获得一些情况要比全力以赴处理积压问题更有效率。此外，你最好去了解一下团队所面对的重大问题，这将会促使你较早找寻出解决之道。许多问题还会在日常工作中冒出来并迫使你持续转移注意力。可以通过