

China-Dub.com
下载

178

这里所谈的开发,包括的以下的实际行动:执行计划、进度安排、程序撰写以及品质验证,而不是讨论什么人做什么工作,在团队中每一个人都必须参与开发的活动,因此,每一位都是开发者。

人们使用软件开发(software development)来表示软件产生的过程。这一点很有意思,这表示软件的形成是一种渐渐成熟的过程,经过一组有顺序而相互影响的步骤演变而成。究竟是什么因素推动软件趋向成熟呢?是团队的创意凝聚。从众多个人的智能产物,逐渐组成一套结构紧密而完整的智能财产,那就是将要推出的软件产品。这又使我想起了本书一再强调的主题:软件开发的基本活动就是将一群个人的智能结合成一项智能财产。

软件开发的终点事实上是开发人员 脑力的极限。



脑力的密切结合

微软团队 — 成功秘诀 China-Dub.com

下载

180

但软件开发的过程牵涉到很多层面。每一个人、团体 互动和科技,三者都在发展,并不是固定不动的。所以, 软件开发没有明显的起点或终点,而是很多人在很多不同 的层面切入,共同创造。虽然每一次的新版,都可以说是 一个里程碑,但是不能把它当成终点,软件开发的终点事 实上是开发人员(脑力的极限)。

很多专家以工程的眼光分析软件开发的问题,而软件 开发的问题也常常被当成是工程问题。但我却常把软件开 发当成社会学或文化现象。和一群人共同创造智能财产, 比起工程上的设计方法论或其他的基本理论要困难得太多 了,在工程上的情况都是有限的,但在软件开发过程中的 问题却是无限的,而且没有绝对的是与非,很多问题都是 要靠沟通和取舍来解决。聪明才智是创造软件的绝对必要 条件,但光有聪明才智并不够。

处于布局时期的"开发",就像解联立方程组一样, 方程式中的变量有无法预测的时间排序和不明确的产品内容。要知道,在该说的全说完但还没真正动手之前,大概 没有人知道软件完成时的模样。在这个阶段,所谓的产品 功能特色只是个描述性的名称,和大概期望的功能,而设 计通常是到了新版产品出炉了才算真正结束。换句话说,

在这个联立方程组中的变量都是不确定的值,而且还会随着开发阶段的演变而改变,这是个极为复杂的问题。

与其说软件开发是一组结构巧妙的 程序,不如说是开发人员脑力的密 切结合。



另一个在布局时期所面临的难解方程式是人的创意, 也就是说你必须在此时就估计出每个组员创造力的效用程 度。我们不仅要在本阶段仔细思考功能特色的内涵,还得 预测它如何能变成具体明确的工作项目,何时能放到组员 的手中加工,并且完成它。

上述的不确定性因素实在是太多太复杂了,以致安排 进度几乎是不可能的任务。当然啦,你还是要设定项目的 整体目标,然后集中精神在第一个里程碑,也许是项目开 始后的一两个月,这大概是你在布局时期时最多所能做到 的了。在第二篇孕育阶段中我们会更详细地谈设定里程碑 的技巧;这是一个让团队练习在短时间内做出具体成果的 好机会。注意每个里程碑的间隔不要超过三个月。

微软团队 — 成功秘诀

一般而言,软件开发比较像是在赶进度,而不像是在演奏交响乐;交响乐是和谐有序而优雅的,而软件开发却是一堆排山倒海、蜂拥而至的工作。这个比喻太可怕了,把它当作是即兴演奏吧!你必须自己知道什么时候该出来表现,什么时候退后一点,大家都是在动态的情况下,和很多人密切合作,任何两个音符都不要互相抵触,让整体表现出的是一段优美的旋律。在一切都不确定的情况下把整个活动带到最高潮,使其美妙的程度不下于交响乐,而且事实上,它本身就是一种喜乐。



## 法则 25

Don't accept dictation 拒绝不合理的命令 China-Dub.com 下载

184

我很惊讶居然有软件开发团队接受非专业人士的指挥,尤其是有关进度的事情。在现代科技领域的世界中竟然使用典型旧社会的方法,简直是大开倒车。有些公司竟然让根本不懂软件开发的人来管理最重要的事 时间、特色和资源是软件开发的金三角 简直是疯了。那些在上位者或是行销人员是假装无所不能的妖魔鬼怪,变个魔术就能拟定进度了。更糟的是,有些专业人员竟然也接受这种缺乏见识、毫无逻辑的领导,还把这种事情当作是标准的作业程序呢。

我曾经接触过数以打计的开发团队,据我私下估计, 大约有三到四成经常为不合理的重大命令所苦。合理的做 法是由负责做事的人来估计时间。当然啦,如果精确不是 目标之一的话,任何人都可以随便乱估。



如果在上位者不让真正执行任务的 人来估计所需的进度,那就是专制。



有些时候,没有好好估计时间是团队自己的问题,是 开发人员和项目经理没有担负起决定达成目标所需时间的

185

责任。把估时间的责任从执行工作者手上拿走,是最最最 反授权的做法。在任何情况下接受这种做法都是极槽糕的, 一开始进度就是假的, 团队如何会有工作能力呢?

没错,我们很能同情那些上位者需要感到自己能控制能预测,但究竟是什么原因使这种荒谬的决策一再上演,而使软件开发永远是个大灾难?

不管是人或组织,大部分的时候都无法从错误中学习。 人有时候很粗心大意,上次做错的事,我试着再做一次能不能做好,完全没有细细思量上回弄错的原因,甚至不去想想看成功软件开发事例给我什么样启示。

这种盲目的现象特别容易发生在进度严重落后的团队中,很多开发人员再也受不了这种死亡进行曲,只好挂冠求去。这些离去的人对于工作要求都很高,才会不容许自己继续和这些官僚大爷们混下去,而且通常都是最重要的人。而留下来的人在怨声载道中,被弃于险地而无人理会。魔鬼在办公室的每一个角落出现。行销人员像个傻子,他们的承诺像是空气,不知不觉流露出愤世嫉俗的态度;高阶管理者充满困惑、窘困和愤怒;顾客觉得自己被骗了。灾难于是开始……



## 要知道,在疯人院中,头脑清醒的人反而会被当成疯子。



慢慢地,云消雾散,大家又重新燃起一线希望,也加入新的成员,新的(或是被原谅的)管理者来带领大家,而新的技术也吸引了开发人员再度投入热情.....上过当却学不了乖的高阶领导者又开始了错误决定,下令软件必须在某月某日做出来,于是灾难的循环又再度开始.....

如果是你,该如何对付这种荒谬的情况呢?如果你 发现自己身陷在这种组织当中,该如何面对困境?记 住!在这种荒谬的环境中,任何正常人都会失去理智, 而组织中弥漫着不理性的行为,它的未来必定走向不可 理谕,你在这里一点希望都没有,身为惟一头脑清楚的 人,必定会被旁人当成是疯子。也许你可以明哲保身地 在这种自我毁灭的组织中勉强活下去,但你不可能会有 什么像样的作为。

但是无论如何总得解决这个问题。所以你在接受不 合理命令的同时,还是要很技巧地提醒你的昏君,这样

187

做是不对的,无法这样做是因为事实的要求。权力不是 万能的,他们的做法行不通,你得帮助你的昏君明白, 大家都在拼命完成皇上的目标,而不是团体的目标,如 果能够多点参与,情况一定会比较好,人们也会因此而 做得更快,更愉快。

开发进度表应该由下而上来拟定,每一个人负责自己的工作,也负责设定它的时间表,负责准时完成工作。责任和充分授权是一体的两面,二者兼备才能拟出合理的开发计划。

## 法则 26

Now go play 把工作当作游戏吧

微软团队·成功秘诀 法则26 把工作当作游戏吧——

189

如果把软件开发当成是一种工作,那实在是一个索 然无趣的职业;从另一个角度来看,我建议你把它当成 游戏。



在布局时期,每件事情都按步就班 进行并不是成功的保证。



如果你引用我的法则来建立你的团队,虽然软件还是同一个,但你会发现他们更自动自发、反应更快、更多幽默、更容易做出令人欣赏的软件。把竞赛当成游戏,你比较容易赢。计算机其实是个充满趣味和挑战的玩具,而开发人员呢?当然是大玩家啰。

在健康的团队中,以游戏的方式来做事是很自然的。 管理者不需要做什么特别的举动,只要避免限制组员的发挥,和注意大家是否乐在工作又自动自发就行了。鼓励组员快乐,不要怕他们分心,这可能是创意的泉源。

软件开发这场游戏,会自然而然地产生规则,做得好的人自然会获得奖励,而做得不好的人自然会受到惩罚。 管理者不必操心正义和仲裁,游戏本身的奖惩比你做的奖 ■微软团队 ——成功秘诀

190

惩要更有效果。



用游戏的心情工作并不是罪过

微软团队·成功秘诀 法则26 把工作当作游戏吧\_\_\_

191

这个游戏能给你什么好处?多玩玩它,练习它,感受它的趣味,并且喜欢它。如此一来,软件创造本身的趣味会使僵硬、骄横和严肃都消失,这样最好。

在布局时期,每件事情都按步就班进行并不是成功的保证。本篇的目的是告诉你如何在发挥团队最大潜能之前,准备好你的团队。本篇并不是成功的公式,而是伟大的先决条件。如果每件事都能在布局时期就照本书的法则进行,那么以后仍然会如此进行,协助你的团队成长起来。