

# MSF 模型在企业研发流程中的应用

● 陈思恩 刘震宇 张 耕

**摘要:**针对国内 IT 企业百人左右的大中型产品或团队在项目管理上存在的问题,文章提出了在企业研发流程中引入 MSF(Microsoft Solutions Framework)软件开发模型,并结合企业的具体情况进行裁减,其能够极大的提高软件工程过程管理能力,从而提高整个软件产品的质量。

**关键词:**项目管理 MSF 软件过程管理

中国绝大多数的软件、互联网企业在 100 人规模以下,开发团队几十人左右,工作经验 3 年~9 年,基本处于原始手工作坊式的刀耕火种阶段,工业化的软件研发流程和方法往往不执行,或执行不得力。本文引入微软公司的 MSF(Microsoft Solutions Framework)软件开发模型,通过 A 企业的实践证明,运用 MSF 框架的软件工程过程管理能大大提高小型软件开发公司的软件研发能力,软件的交付期也得到了基本保证。

## 一、MSF 基本模型

1. 组队模型。MSF 把软件开发分成了六个小组,分别是:程序管理组、产品管理组、开发组、用户培训组、测试组、安装管理组。组队的原则是小队(一般 3 人~8 人)、多侧面;角色交叉、目标一致;人员技术、业务精;关注能力和交货期;对项目的前景认识一致;人人参与设计;善于总结经验;共同管理、共同决策,项目人员同地工作等。

2. 过程模型。MSF 过程模型主要确立了四个重要的里程碑:前景范围确认、项目规划确认、开发完成、对外发布,通过控制这四个里程碑来分解管理项目过程。

3. 应用模型。MSF 应用模型是分层次的应用模型,大体可分为三层,用户层、业务层和数据层,各层次通过标准组件进行封装,互相通讯调用来完成系统任务。

4. 风险模型。MSF 风险管理过程主要包括:风险识别、风险表述,通过分析、计划、跟踪和控制过程,最终解除风险。

## 二、MSF 在 A 企业呼叫中心项目中的具体应用

本文以 A 企业呼叫中心项目为例(合计开发人员 50 人),结合该项目中的数据资料,对此项目中服务商的选择进行具体的分析和研究。

1. 组队模型裁减。首先,确立项目经理角色,符合一般企业的管理模式,比较容易被接受。如果有多人同时负责的话,容易产生责权理不清楚,互相扯皮的现象。有一个项目经理对项目完全负责,遇到问题容易很快得到解决;他作为项目组代表,负责向上级汇报工作,能使其他人全力投入到项目中,而不至于在日常的事务中耽误太多时间,从而在某种程度上也提高了工作效率。

产品管理经理(以下简称产品经理)是项目中非常重

要的角色,他可以对技术不是很精通,但是必须对产品所服务的领域非常熟悉,最好是领域专家,在他的带领下,项目才不至于偏离预先设定的前景范围。他必须对产品的需求能做出很好的把握,在适当的时候能进行流程重组,对产品的可用性和易用性有最终决定权。通过设定产品经理,主要的感觉是产品受用户的欢迎程度增加了,无用的特性少了,因而也更容易成功。

2. 开发过程管理。在过程模型上,主要包括四个重要里程碑:①前景/范围确认;②项目规划确认;③开发完成;④对外发布。

A 企业把 MSF 的各个阶段对应到传统的项目开发各阶段,目的是使企业所有人员便于理解和使用。其中“前景范围确认”对应传统的“可行性分析”;“项目规划确认”对应“需求分析”和“项目计划”;“首次运行”对应“开发完成”,“发布”的意思和传统基本相同。同时,A 企业根据企业的具体情况对流程进行了相应调整,把整个流程分为可行性分析、需求分析、开发计划、开发过程和结项总结五个阶段,下面分别进行说明。

(1)可行性分析。在做可行性分析的时候,要充分考虑企业以前的各种技术和市场积累,还有目前的资源可用性情况,特别是要做好风险分析。通常企业在项目立项前没有充分考虑各种情况,认为这个项目比较简单,应该没什么问题,结果是没有做得很成功,进度上也拖了一段时间。在后来结项分析的时候,认为主要的问题就是领域的区别造成了企业内部没有人对该领域特别熟悉,缺乏领域专家,并对上述风险估计不足,也没有对风险进行较好的管理,所以造成了项目的不成功。

上面提到,可行性分析一般是由未来的项目经理和产品经理完成,必要时还需要市场人员的参与,项目经理主要考虑技术可行性,包括项目最初估计的进度表和资源需求情况;产品经理主要考虑市场和经济上的可行性(主要是针对软件产品而言)。只有预先对各种问题进行完备的分析后,才能得出正确的决策。不要到后来因为那些事先没考虑到的,但应该想到的各种原因造成项目失败;或者虽然完成了,但是没有取得预期的效果,不能给企业带来较好的收益。

只有在可行性分析通过评审,企业高层领导者认可的情况下才能付诸实施。通过可行性分析,揭示了即将面临的各种问题及风险,使得企业内部对该项目有了一致的认识,在后来的资源申请上也更容易得到高层支持,更易于导致项目成功。

(2)需求分析。需求管理是软件开发中非常重要的部分,在一般的MIS型项目中,准确的把握需求往往是项目成功的关键。但需求管理也是个困难的过程,太多项目的需求都没有良好的管理过程,往往导致项目后期的大量修改或者直接使项目失败。

需求的管理主要由产品经理负责,其中最终用户的实时参与是一个非常重要的因素。在需求采集阶段,A企业主要采用了原型法,使用Visual Studio.Net建立最终产品的界面,然后把功能实现和界面一一对应起来,和用户进行讨论,并不断的修改界面。最终在基本达成一致后,对应原型写出需求规格说明书,在评审后纳入基线管理。

在需求的制定过程中,产品经理必须和项目经理、开发人员、测试人员进行良好的沟通,使项目组全体都参与到需求分析中来,并共同确定需求的关键特性:①项目的范围:在需求分析中,首先必须明确项目的范围,去掉那些看似属于该项目其实不该在项目中的需求特性。特别是在一些MIS项目中,客户往往把一些属于他们的日常工作但不属于该项目的需求提交给项目组,这时就必须分清项目的范围,不要在项目中加入太多不应该做的东西,否则往往会导致项目范围无限扩大,最终只能是使项目失败。②需求的优先级:需求的优先级是非常重要的特性,只有在准确把握的需求优先级的基础上才可能规划外部里程碑(产品版本)和内部里程碑(开发的阶段性,后面会讲到)。通常是用户最关心,使用最频繁的功能应该属于高优先级,而那些不怎么重要或很少用到的功能应该属于低优先级。A企业必须在产品的开始版本和项目的开始就把重点放在高优先级的需求上,而对于低优先级的功能可以在项目后期根据需要进行裁减或纳入下一个版本规划。③产品的易用性:产品的易用性反映在原型中,是原型法的一个非常重要的作用。很多产品的失败其中一个重要原因就是易用性比较差,虽然它在功能上满足了用户需求,甚至可以说功能很强大。通过原型法,能让用户看到并模拟使用最终的产品界面,能在需求阶段通过修正软件界面来适应用户的偏好,从而在很大程度上提高了产品的易用性,使项目更容易成功。④其他需求特性:如性能要求、健壮性等。这些特性是产品的非功能性需求,也是项目成功的关键因素,特别是在一些大型的涉及重要领域的管理信息系统中。

需求分析完成后,项目组需要对项目的初步计划进行重新审定,一般都需要变更项目时间表和资源需求。需求分析的完成也意味着项目其他部分可以齐头并进,如概要设计、测试计划、用户说明书,这也在某个方面证明了需求分析的重要性——它是下面所有活动的基础和准绳。

(3)开发计划。软件开发中的计划性是非常重要的,一

个没有良好计划的开发项目能够成功的机会非常小,除非有天才的程序员再加上好运气。开发计划的主要内容包括:项目进度安排、人力资源安排、风险管理策略等。

项目的进度安排和人力资源安排可能是开发计划中最重要的部分,也是最难以估计的部分。一般国内的中小软件企业对项目工作量和开发人员能力的量化程度不高,所以导致进度和资源安排不确切,有时候甚至是相差很远。目前一个最实际的办法就是根据以往项目的积累,但必须要求是同一领域的类似项目,这样才有较强的可比性。由于这些计划安排是预估粗略的,所以还必须在以后的项目各阶段完成后进行合理的变更,反应项目的实际需求。微软的办法是把进度估计的权限交给开发人员,由开发人员根据自己的经验进行估计,由于一般开发人员往往会高估自己的能力,估计的进度也会相应偏短,最后再做适当的延长。这种办法有它合理的地方,在中国还需进行实践摸索。

对于进度的估计业界有个经验公式,即最初预估的时间再乘以2.5,可能是最后的完成时间。因为许多人在估计进度的时候,往往忽略了很多非开发时间,如与客户沟通的时间、项目组沟通时间、企业培训时间、假期等,所以在估计进度的时候,一定要全方位周全考虑,在尽可能的情况下宁愿把进度估计的长一点,免得在项目后期导致非常被动的局面。后面将具体讲到采取的阶段性的开发方法,这种方法的运用反映在进度估计时必须在各阶段间预留缓冲时间,以解决那些事先没有预料到的活动。如果进度表和要求的出货时间有冲突,宁愿砍掉一些不重要的功能,也不要盲目增加人手,这种做法可能会导致产品质量下降,最终得不偿失。

(4)开发过程。在项目的开发过程中,A企业采用了阶段式的开发过程,这也是微软企业所推荐的开发过程。在开发过程的初期,首要的活动是概要设计。概要设计的目标是简单、适用、能够覆盖所有的需求并能支持后面的阶段式开发。微软的应用方案解决模型是基于服务的三层(多层)架构,包括用户层,业务层和数据层,各层之间采用标准的接口进行通讯。

阶段开发过程不是传统的根据模块划分来依次完成各模块,最后再进行项目的整合,而是在每个阶段完成后,项目都可以推出产品,只不过该产品的功能比最终产品的功能弱一些。阶段性完成项目比传统的开发方法最明显的优点是不必到项目的末期才开始整合产品,使产品模块之间协作产生的问题及早产生,也及早修正,从而项目的风险也大大减小。当然采用阶段性开发方法也有相应的代价,最大的代价可能是反复的整合、测试已经完成的模块,但采用相应的一些自动化工具可以减小这个代价。

一般在开始的阶段进行的是系统架构和最重要的功能,后面的阶段是相对不怎么重要的功能。这样的分配有利于最终用户在早期就能看到系统的大致模样,便于他们及早的对产品提出意见,并对相应的错误进行修改;也有利于项目组在项目后期时间很紧的情况下,去掉一些不重

要的功能,把它们纳入下一个版本处理,确保产品的推出时间。迭代的顺利进行依赖于良好的架构设计,前面阶段的设计应该给后面要加入的功能预留出各种接口,并能使后面的工作在前面的基础上继续进行下去。

这种在开发阶段的迭代方式不同于整个项目的完全迭代开发,后者是项目的需求、概要设计、开发等全部是迭代进行,一次迭代要进行所有的项目活动。至于谁优谁劣可能在不同的情况下有不同的说法,需要根据项目和自身的情况合理采用。还有就是迭代的次数也要根据项目的具体情况而定。不能太多,导致重复的工作量过大;也不能太少,使得该方法退化到传统方法。通常把项目分了四个阶段:架构完成、主要功能完成、其他功能完成、整合发行。实践证明,这样的实施比传统方法确实在很大程度上减小了项目失败的风险,再没有产生那种“似乎永远也做不完的感觉”。

(5) 结项总结。很多企业在项目完成后往往忽视了最后的总结,没有把在上个项目中得到的经验教训进行分析,转化成企业的巨大财富。项目的总结是整个项目不可缺少的重要组成部分,只有通过详尽的充分的项目总结,才能使项目组的所有成员对项目的历程有一个清楚的了解,提高他们对软件项目的认识。

A 企业的做法是在项目完成后首先由各个项目成员写出各自的总结报告,包括所从事的工作、任务的完成情况、遇到的问题及解决方案、对项目过程的意见和自己的想法等内容。项目负责人要把整个的项目历程整理成一份文件,其中包括项目的介绍、项目进行的具体资料(如实际花费时间、源代码数、功能模块数量等)、项目计划与实际的比较等。

在上述完成后,全体项目参与人员举行项目结项工作会议,对各人所列举的问题及想法进行讨论,目的是得出好的经验教训,从而指导后面项目过程。会议可由分别针对的问题分为几个部分,如项目过程方面的、质量管理方面的、技术方面的等,整合后形成结项会议报告。

项目负责人最后把项目历程、资料、在结项会议中总结的经验教训等整理成一份总的的项目过程文件,归档并分发到各成员和上层领导,并由项目经理向上层领导汇报,这时,一个完整的项目才真正告一段落。这些项目资料给以后的项目提供很好的模板和借鉴意义,并可以作为以后项目预估的依据。

3. 风险管理。微软企业认为,软件开发是一个风险驱动的过程,由此可看出风险管理在软件项目中的重要性。一个项目的风险有许多来源,如客户、进度、开发过程、人力资源等,忽视风险的后果可能是成本超支、进度推后,最严重导致项目失败。

MSF 的风险管理原则是:①风险应该在整个项目的进程中一直被估计,并且作为项目决策的依据之一。②有效的风险管理过程覆盖了所有关键的人力、过程、商务及技术领域。③风险在纳入管理前必须被清晰的表述。④重要的风险必须优先被处理。

MSF 风险管理过程包括以下阶段:风险识别、风险陈述、风险分析、处理计划、风险跟踪、风险控制、风险解除。

在中小企业的风险管理过程中,一般项目经理担任风险管理角色的角色,但同时需要另外的资深开发人员辅助,一起完成风险管理的任务。他们负责维护十大风险清单(不一定非要列出十个),并在项目进程中随时对风险清单进行更新。对风险的评级 MSF 采用的方式是:风险影响程度=风险的可能性×风险发生造成的损失,根据风险影响程度的大小对风险进行评级。

在项目实施中,A 企业总结的一些高风险事件主要有:需求的不准确、项目时间表过于短促、开发一个从前没进入的领域软件、开发人员对工具的不熟悉、人员流动频繁、使用了外部软件中间件等。如果对这些风险不提前做出计划,可能会对项目的顺利进行造成极大的破坏,甚至直接导致项目失败。针对每一个风险,A 企业需要列出 who, when, how, how much 等事项,并对风险处理的结果进行追踪,最后决定是否已经解除风险或再进入风险处理循环。

### 三、总结

引入 MSF 管理思想后,项目的成功率比原来增大了很多。企业研发流程必须尽可能 MSF 模型化,工具化,没有银弹。所有的一切都是在一个公共的工具平台上,至少需要需求管理工具、版本控制工具、日志构建工具、Bug 管理工具等。德鲁克认为,管理既不是科学,也不是艺术,管理是实践,正如软件工程的世界级大师 Ivar Jacobson 所说,“我相信没有一种技术、没有一种流程能够解决所有的问题,我更相信基于实践的方式。你需要用不同来源的实践,结合自己的流程,改进自己的流程和工作方式,这样才能立于不败之地。”

### 参考文献:

1. (美)迈克尔·科索马罗 理查德·赛尔比 著,程化译. 微软的秘密. 北京:北京大学出版社,西蒙与舒斯特国际出版公司,1996.
2. (美)史蒂夫·麦克康奈尔 著,余孟学译. 微软项目求生法则. 北京:机械工业出版社,2000.
3. (美)Karl E. Wiegers 著,陆丽娜,王忠民,王志敏译. 软件需求. 北京:机械工业出版社,2000.
4. (美)卡耐基梅隆大学软件工程研究所著,刘孟仁译. 能力成熟度模型(CMM) 软件过程改进指南. 北京:电子工业出版社,2001.
5. (美)Frederick P. Brooks Jr. 著,人月神话(影印版) The Mythical Man-Month. 北京:中国电力出版社,2003.
6. 匡晓红. 基于 MSF 的风险管理模型. 中国科技信息, 2007 (17).

基金项目:福建省社会科学规划一般项目(项目号:2011B223)。

作者简介:刘震宇,厦门大学管理学院教授、博士生导师,陈思恩,厦门大学管理学院博士生,高级工程师,张耕,厦门大学国际经济与贸易系副教授。

收稿日期:2012-10-26。