项目管理指南

Ver: 0.1(Draft)

**目 录**

[第一章 前言 4](#_Toc289170571)

[第二章 基本原则 4](#_Toc289170572)

[第三章 生命周期(过程)模型 5](#_Toc289170573)

[生命周期模型 5](#_Toc289170574)

[生命周期最佳实践 6](#_Toc289170575)

[每日构建 6](#_Toc289170576)

[流程整合 7](#_Toc289170577)

[持续集成 7](#_Toc289170578)

[可交付成果 8](#_Toc289170579)

[启动过程 8](#_Toc289170580)

[规划过程 8](#_Toc289170581)

[实施过程 8](#_Toc289170582)

[监控过程 9](#_Toc289170583)

[收尾过程 9](#_Toc289170584)

[第四章 组队模型 9](#_Toc289170585)

[项目领导团队 10](#_Toc289170586)

[变更控制委员会(CCB) 10](#_Toc289170587)

[执行团队模型 11](#_Toc289170588)

[团队模型最佳实践 13](#_Toc289170589)

[比例缩放MSF团队模型 13](#_Toc289170590)

[功能小组模型 14](#_Toc289170591)

[第五章 项目管理过程 15](#_Toc289170592)

[立项流程 15](#_Toc289170593)

[变更管理 16](#_Toc289170594)

[项目收尾 16](#_Toc289170595)

[第六章 其他内容 17](#_Toc289170596)

[项目文档编号规范 17](#_Toc289170597)

[项目工作环境 17](#_Toc289170598)

[第七章 附录 18](#_Toc289170599)

[术语表 18](#_Toc289170600)

[进一步资料 18](#_Toc289170601)

[生命周期概览 19](#_Toc289170602)

[常见问题解答 20](#_Toc289170603)

[参考文献 21](#_Toc289170604)

[修改记录 21](#_Toc289170605)

# 前言

本指南是项目基地实施各类项目的标准指南。依据MSF1、敏捷等流行软件过程(方法)，参考PMBOK(2008版)，结合公司开发管理经验编纂而成。

本指南力求实效，以项目管理最佳实践和核心原则作为核心集，以项目团队成功完成项目所必需的管理知识作为主体，实践优先，理论居后。

本指南包括基本原则、过程模型、组队模型、项目管理过程及附录组成。

本指南自身具备自我演进特性，项目基地与教学部门定期结合项目实践经验与项目管理理论研究成果对本指南进行修订。

本指南的读者包括项目基地专/兼职项目经理、开发、测试、用户体验、产品管理等。

备注：

MSF 除非特别说明，本指南中提到的MSF版本为3.0。在本指南编纂时MSF的最新版本为5.0，分为For CMMI和For Scrum两个分支。

# 基本原则

* 愿景优先 项目团队必须与客户保持良好合作，并必须就项目的愿景达成一致。愿景是指向项目成功的第一要素。
* 交付加值 项目团队与客户一起紧密合作，为客户提交有价值的项目成果。
* 每日构建 无论采取何种软件过程，项目团队必须保持以每日构建为基础的团队协作模式。
* 持续集成 项目实施过程中，保持以短周期，持续集成，不断推出可用版本。
* 平等协作 项目成员之间只有工作职责不同，无高低贵贱之别，团队的最主要目的是与客户一起实现项目愿景。
* 适应变化 项目是一个逐步细化，逐步清晰的过程，在项目初期项目团队所能获得的材料有限，当项目团队遇到需求变化时应该采取与客户一起面对、共同解决的态度，高质量的完成项目目标，实现项目愿景。
* 开放透明 项目团队应保持开放透明，项目团队应制定沟通计划，定期与客户沟通项目情况，并欢迎客户加入到项目团队中共同工作。
* 持续改进 项目组应该在每个迭代/增量周期进行总结，不断积累、提炼项目管理及技术经验。

# 生命周期(过程)模型

本指南鼓励采用增量、增量+迭代为基础的软件生命周期模型，反对采用瀑布式生命周期模型。

每日构建、持续集成是本指南要求必须遵守的质量保障手段。

## 生命周期模型

从经验上来看，与客户签订的合同一般都是总价类型的合同。这类合同的特点是，价格固定，产品范围基本确定。操作过程中一般都是合同与需求收集同步(或基本同步，甚至需求先行)进行，合同签订时附带“最终需求以双方签订的需求确认书为准”的条款。从生命周期角度看，已经将生命周期划分为需求阶段、实施阶段、验收三个大的阶段。因为产品范围基本确定，项目团队在选择项目模型时可以优先选择增量、迭代+增量的生命周期模式。并将每个迭代作为一个小的里程碑，每个里程碑完成几个完整的功能/模块，经过内部测试、集成后，邀请客户核实。通过持续不断的交付可用的产品来保持项目节奏，把握质量。

因为软件项目的不确定性，往往是前期需求调研时候确定的需求并不完整，或者因为害怕后续繁琐的变更手续导致客户需求膨胀(客户什么都要，而不是以最终交付价值为导向)，在需求收集过程中应加强对客户的引导，以交付最终为导向，在共同愿景前提下欢迎变化。

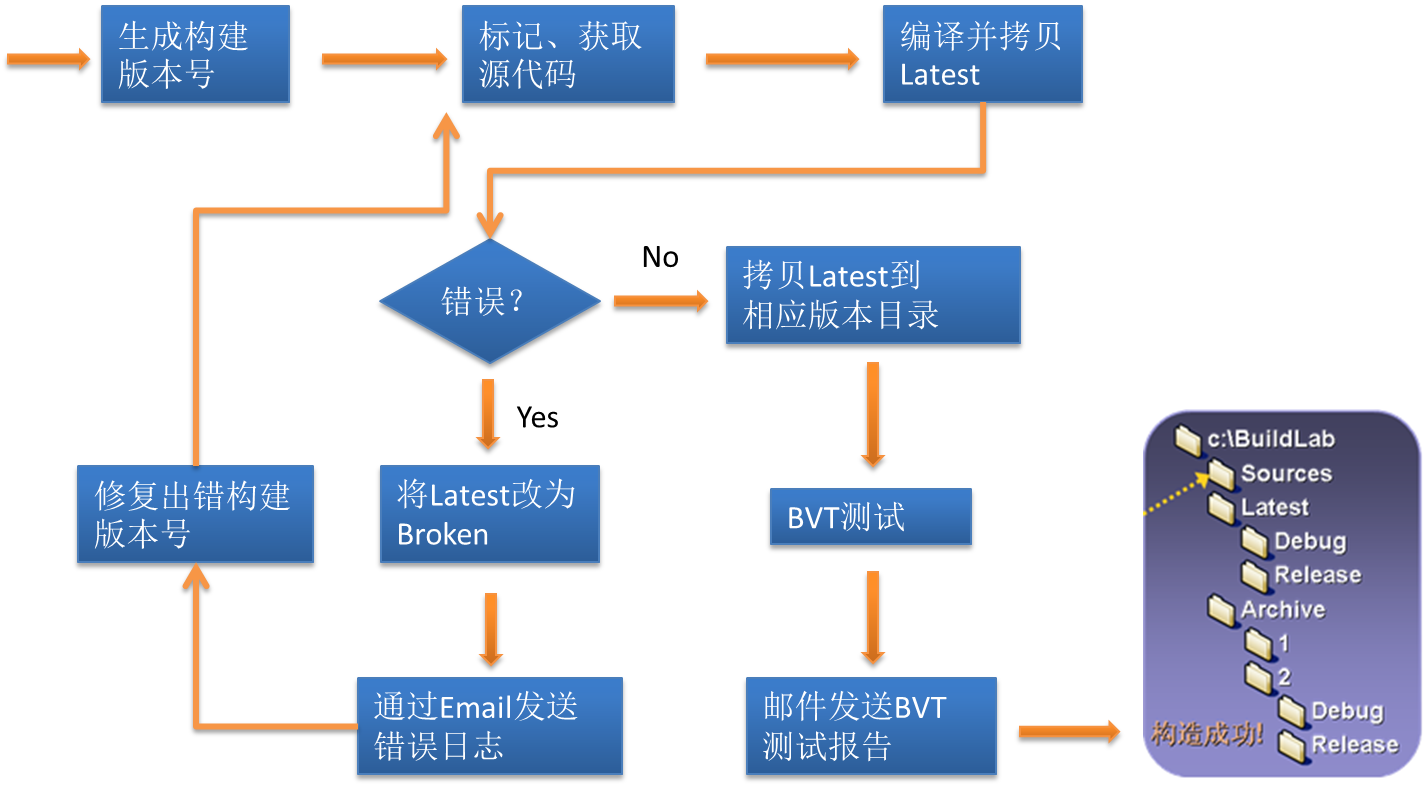
内部产品项目及创新性较强的项目，推荐选择迭代、迭代+增量的生命周期模型。实施时可以只了解部分需求，在每个迭代中完成一部分功能，不断验证完善产品创意，通过持续集成来确保质量。根据迭代周期的长短，可以划分为一些阶段。如计划阶段、设计阶段、实现阶段、稳固阶段、发布阶段等。

|  |  |
| --- | --- |
| D:\YuanXu\My Documents\My Pictures\tmp\msf.png  一个完整迭代 | 迭代式项目推进 |

## 生命周期最佳实践

每日构建、持续集成是本指南要求必须采用的质量保障手段。无论项目团队采用何种项目生命周期，这两个最佳实践都可以有效推动项目，持续不断地为客户交付价值。

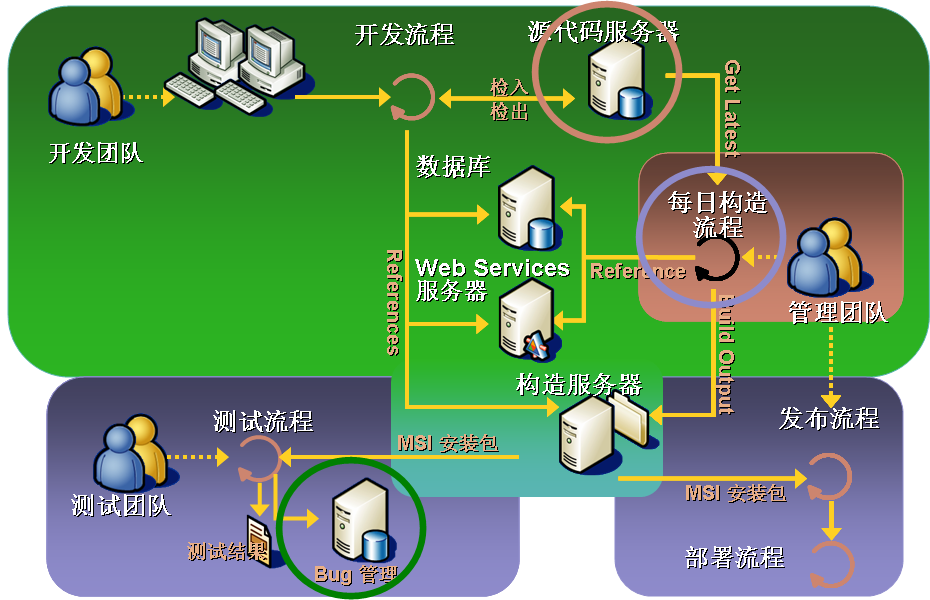
### 每日构建



实施要点：

* 每日构建工作一般由测试团队负责，并最好能够实现自动化。
* 每日构建要与BVT测试结合，每个成功的Build都应该通过BVT测试。

### 流程整合



实施要点：

* 开发团队负责单元测试
* 测试团队管理测试环境和构建版本
* 开发团队与测试团队依靠Bug库和每日构建流程协作
* 管理团队(PM组)更关注每日构建情况

### 持续集成

当系统由多个子系统/模块组成时，在每日构建的基础上，要不断进行集成。集成一般需要开发人员配合，并且一般不是每日发生。一旦模块集成完毕应该并入每日构建和BVT测试的内容。

## 可交付成果

在可交付成果部分，本指南根据PMBOK的思想从宏观角度分为启动、规划、实施、监控，收尾五个过程组。在这个几个过程组中规定必须完成的几个可交付成果来指导项目管理，项目团队可以根据实际情况增加其他的可交付成果。

注意：

1、PMBOK划分的项目生命周期，属于顺序型生命周期(瀑布模型)，对于软件项目的宏观角度适用，但不应该直接作为软件项目的生命周期模型。

2、如果未采用本指南中推荐的技术架构，项目团队需要安排“选定技术框架”里程碑。

### 启动过程

* **可交付成果** 项目章程、干系人登记册
* **里程碑** 项目章程被批准

### 规划过程

* **可交付成果** 项目文件（项目规划、需求规格说明、WBS、风险列表、测试计划）；
* **里程碑** 项目规划被批准

### 实施过程

* **可交付成果** 项目文件（WBS（更新）、测试计划（更新）、测试用例）；产品文件（\*设计文档，\*源代码、\*编译后版本、自动化测试脚本）
* **里程碑** 范围完成

### 监控过程

* **可交付成果** Bug趋势图，项目进度图(燃尽图)，范围核准单

提示：

Bug趋势图和项目进度图，应该每天提交项目的主要干系人

### 收尾过程

* **可交付成果** 产品文件（\*源代码、手册、\*编译后版本、安装程序及源代码、测试报告）；项目文件（\*验收报告）；组织过程资产（更新）
* **里程碑** 发布

# 组队模型

组队模型规定项目团队的组成和组织结构。本指南推荐的项目组对模型包括项目管理团队、CCB、项目执行团队三个部分。

项目执行团队是实现项目目标的具体负责人(单位，团体)，项目管理团队和CCB是为了顺利推动项目实施的外围保障性组织。项目团队应该根据项目特点决定组队模型。有些项目（如外包型）要求建立至少建立项目执行团队和CCB甚至要包含全部建立三个团队；有些项目（如内部项目）则可能只需要CCB和甚至只需要项目执行团队即可。

领导团队负责制团队模型

## 项目领导团队

项目管理团队是项目的最高领导机构，由客户主管高层领导、项目基地高层领导组成，对项目大的方向、范围、重大变更负责。

项目管理团队成员如能更深入的和项目执行团队（内、外）共同工作，推荐由项目管理团队肩负起CCB的职责，而不是仅仅在重大变更上进行决策。

项目团队可以根据实际情况建议成立项目领导团队。

## 变更控制委员会(CCB)

变更控制委员会是需求规格说明确认以后的重要变更仲裁机构，一般由客户负责人/客户项目经理、项目执行团队负责人/项目经理构成。CCB成员必须是对项目直接负责的人，即能够对变更所引起的项目范围、时间、成本等因素起到决策作用。CCB有权对变更进行跟踪。

参与到大型团队的小团队(特性团队)的CCB由小团队的程序管理、产品管理及上层团队的程序管理、产品管理组成。

提示：变更控制流程请参阅第五章。

## 执行团队模型

本指南推荐采用MSF组队模型，也可以采用Scrum推荐的组队模型。这两个组队模型有很多相似之处，相较而言MSF组队模型更加细致全面，Scrum的团队模型参考附录中的参考文献。

本节摘取自MSF组队模型，关于MSF组队模型的更详细的资料可以参阅附录。

团队模型基于这样一种前提，即任何技术项目都必须达到特定的关键质量目标才能被认为是成功的项目。但，每一个目标的达成，都需要相关的、不同领域的知识和技能的综合运用。把不同领域的知识和技能分别分配到不同的角色中去，每个角色都具有满足项目成功的标准。如果任何一个角色无法实现其目标，这都会危及到整个项目。因此，小组模型中的每一个角色都认为是同等重要的，重要的决定都要共同作出。

*MSF团队模型的特点：*

1. 清晰的责任，共同的职责

MSF 将工作进行中需要共同承担的职责和确保工作如期完成的责任结合起来。子团队中的每个角色都代表了对项目的一种独一无二的观点，但是没有哪个个人能够完全代表所有的不同质量目标。

2. 赋予小组成员权力

在一个高效的团队里，所有的成员都被赋予权力以便根据他们自己的承诺交付任务，并且充分信任团队的其他成员也能实现各自的承诺。MSF团队赋予成员权力以帮助他们履行承诺。

3. 聚焦业务价值

MSF 团队模型提倡将团队决策建立在：（１）团队对客户业务全面的认识（２）项目交付过程中客户能动的参与之上。“产品管理”角色在团队中担当客户的拥护者，这个角色常常由客户组织中的成员来担任。

4. 共同的项目设想

MSF 全力提倡采用一个共同的设想，以便把注意力放在团队的工作方法上。

对项目和过程的目标有一个清晰的了解是很重要的。因为一个共同的设想将使成员对软件的猜测明确化，并确保所有参与者都在为完成相同的目标而努力着。共同的设想是 MSF 团队模型的基础之一。

当所有的参与者都了解了这一设想并朝着这一设想工作的时候，团队便能够根据成员使自己的决策与小组意图相吻合，从而获得他们的权力。

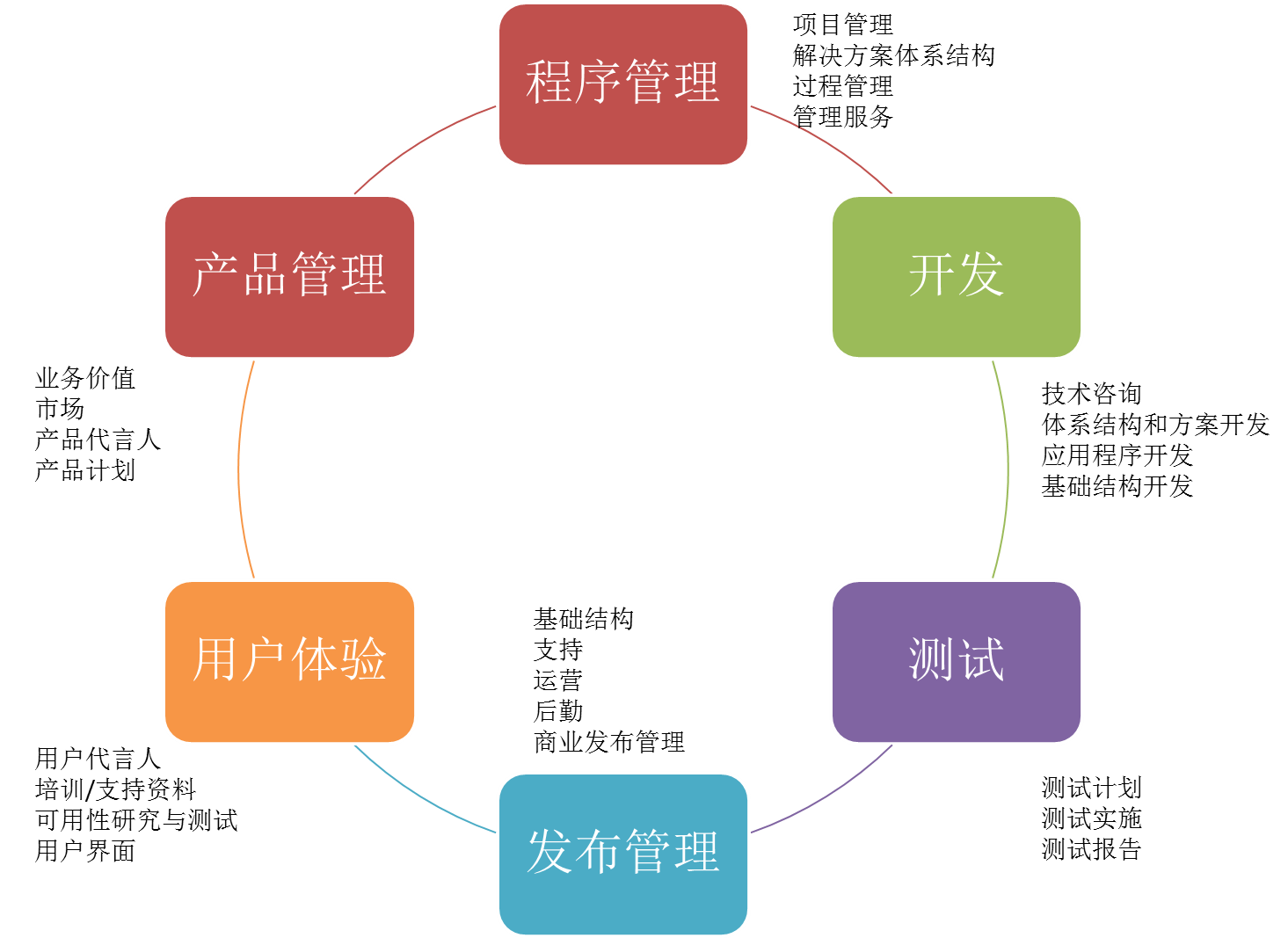
没有共同的设想，团队成员可能出现与目标相抵触的观点，团队成员很难确定自己的成功，因为这种成功依赖于他们评价成功的设想。

5. 保持灵活，预测变化

MSF 认为事情在不断发生变化，将一项 IT 解决方案交付项目与这些变化相孤立是不可能的。

6. 推动开放式沟通

MSF 提出一个开放式的真诚的沟通方式，这种沟通存在于团队内部之间。开放的信息流不仅减少出现误解的频度，而且确保了所有成员可以降低项目周边环境中存在的不确定性



要说明的是，上面给出的ＭＳＦ团队中各角色的职责分工，不表示任何组织机构或工作职位的固定设置。因为ＭＳＦ是一个可伸缩的框架，它仅给出一个团队组织的指导。这些角色应该随着组织的变化，而有所变化。使用ＭＳＦ团队模型，关键在于为了更好地实现项目的目标，清晰地理清角色和它们的职责的分布关系。

### 团队模型最佳实践

1. 组建小型专业化团队（一般不超过１０人）

2. 在同一地点共同工作（团队内部沟通、与客户的沟通都很方便）

3. 要求客户加入项目团队（制定特定接口人）

4. 全体参与项目重要活动（项目不神秘原则）

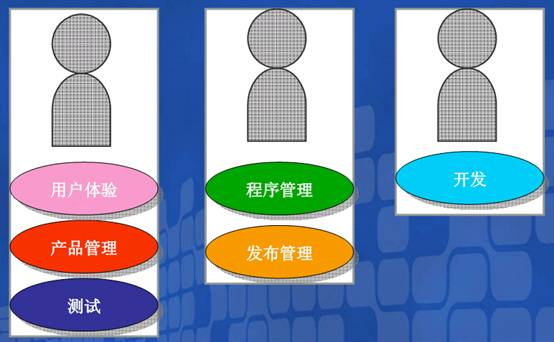
5. 在复杂的项目中，“程序管理”角色分成“项目经理”和“架构师”两种职责。

### 比例缩放MSF团队模型

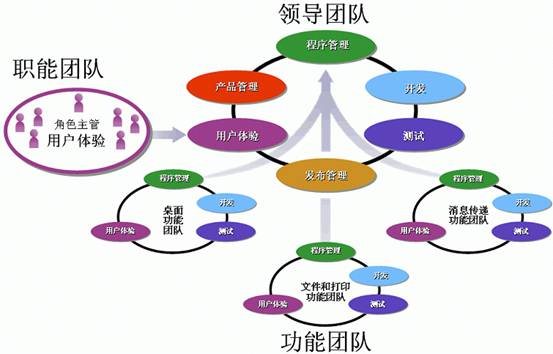
缩放原则：



三人小团队的例子：



### 功能小组模型



# 项目管理过程

## 立项流程

项目可分为内部项目（内部产品开发、技术积累型）和外部项目。外部项目指定项目章程的依据是商业合同；外部项目的依据是商业论证和项目工作说明。

* **内部产品开发立项流程**



任何人都可以作为内部产品开发的发起人。发起人提交商业论证和项目工作说明给管理团队任一成员（接收人），接收人召集管理团队成员召开项目评审会，评审成功后交由项目部实施或安排进入创新基地。

* **技术积累型项目立项流程**

结项时由PM申请提出技术积累建议提交给技术总监，技术总监可根据情况召集技术委员会成员评审或决定启动/不启动项目。

* **外部项目立项流程**

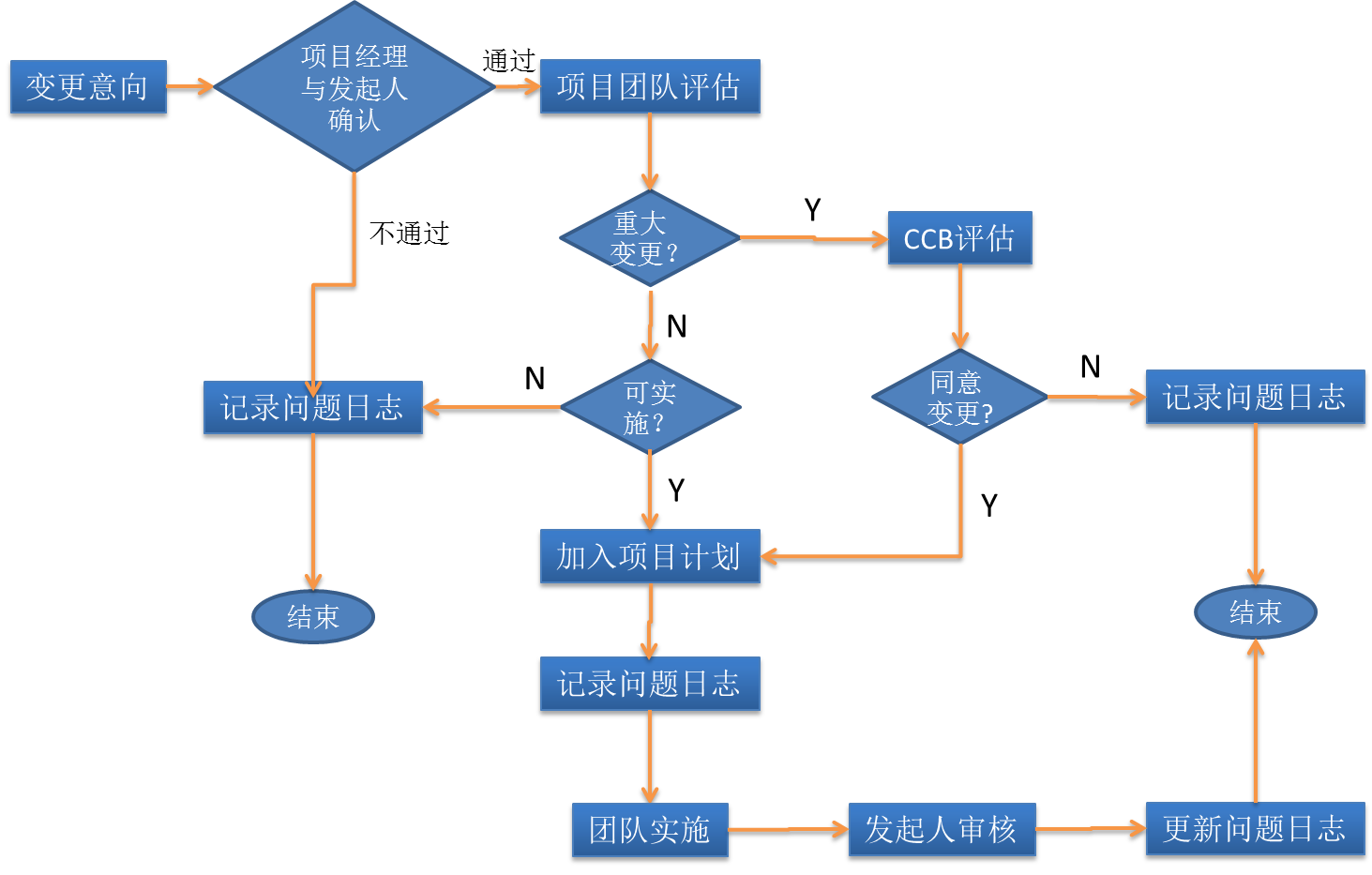
市场部签订的外部合同，项目部必须承接。由项目总监负责协调资源组建项目团队。



## 变更管理

从项目管理的角度来说，任何人都可以提出变更申请。但就软件项目管理实践来说，这里的变更一般都是由客户发起的业务/技术变更。

客户方业务人员提交变更申请给客户方项目管理(或高层业务主管)，客户方项目管理(或高层业务主管)签字后提交项目团队PM，项目团队PM初步评估变更对项目的影响，如果影响较小，则可安排入项目计划，将变更申请归档；如果变更影响较大，则有PM提交CCB，CCB批准后由PM调整排入项目计划；CCB拒绝后记入问题日志。



## 项目收尾

项目收尾过程分为正常收尾和项目异常中止的收尾。正常收尾需要经过客户的验收，项目总结之后更新组织过程资产，流程如下：



异常中止一般是由商务层面无法合作或者技术实现层面无法实现项目目标引起，异常中止需要由项目经理向管理团队(项目总监接口)提交中止项目申请，经管理团队审核后进入内部收尾过程，流程如下：



# 其他内容

## 项目文档编号规范

* 文档编号由固定前缀+项目简称+类别+编号四部分组成；固定前缀为OTQP；教学部主导的创新类项目固定前缀为EDQP。
* 文档类别编号：

01 愿景/范围

02 需求&设计类

03 计划

04 测试

05 变更类

06 报告类

07 风险类

08 部署&发布

09 使用手册

## 项目工作环境

* 项目工作环境包括软件开发环境、项目管理基础环境、测试环境组成。
* 软件开发环境由各项目团队PM统一规定；
* 项目管理基础环境包括项目网站、版本控制系统、Bug库组成、电子邮件；
* 测试环境由项目团队测试组成员负责全权管理。

# 附录

## 术语表

* 项目文件 项目管理过程中产生的文件，如项目章程、项目规划、风险计划、变更单等等
* 产品文件 与项目成果相关的文件。如源代码、安装包等
* 项目团队 广义上是包括项目的所有干系人，项目管理团队、CCB和项目执行团队；狭义上仅仅是指项目执行团队
* 项目执行团队 具体负责实施，完成项目目标的临时团体
* 客户 产品的出资方
* 用户 项目产品的直接使用者，他们有可能是客户方人员，也可能不是客户方人员
* 组织过程资产 包括任何或全部与过程相关的资产，可来自任一或所有参与项目的组织，用于帮助项目成功。这些过程资产包括正式和非正式的计划、政策、程序和指南。过程资产还包括组织的知识库，如经验教训和历史信息。项目团队成员有责任在项目全过程中对组织过程资产进行必要的更新和补充。组织过程资产可分为“流程与程序”、“共享知识”两大类。详情可参阅PMBOK(2008)P.25

## 进一步资料

*OTQP-PM-1管理规范*

OTQP-PM-1-1 项目管理指南

OTQP-PM-1-1 版本管理规范

OTQP-PM-1-2-1 源码规范(C#版)

OTQP-PM-1-2-2 源码规范(Java版)

OTQP-PM-1-2-3 源码规范(C++版)

*OTQP-PMT-2过程模板*

OTQP-PMT-1-1 项目章程模板

OTQP-PMT-1-2 干系人登记册模板

OTQP-PMT-2-1 需求确认书模板

OTQP-PMT-2-2 需求跟踪矩阵模板

OTQP-PMT-3-1 项目规划模板

OTQP-PMT-3-2测试计划模板

WBS模板

风险列表模板

OTQP-PM-2-6 项目进度报告模板

OTQP-PM-2-7 变更跟踪单模板

OTQP-PM-2-8 问题日志模板

OTQP-PM-2-9 验收报告模板

OTQP-PM-2-10 范围/变更核实模板

*项目数字仪表板*

OTQP-PM-0-2 项目进度

OTQP-PM-0-3 Bug趋势

## 生命周期概览

增量式生命周期与顺序(瀑布)式生命周期在需求收集与分析阶段有些相似，但是一般会限制需求收集的时间，然后分成不同的功能团队。每个团队每次开发一个功能，完成测试和集成，再开发另一个功能。

迭代式生命周期在每个迭代收集部分需求，然后进行设计、编码，但是一般对集成不做要求。

迭代+增量式生命周期结合了迭代和增量的优点。但是迭代更短，增量更少。

下面给出对生命周期模型的概览：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 增量式： | 需求收集 | 分析选择整体架构 | 设计、编码、集成和测试 | 设计、编码、集成和测试 | 设计、编码、集成和测试 | 最终集成 | 最终测试 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 迭代型： | 需求收集 | 原型阶段：  分析、设计、编码 | 原型阶段：  分析、设计、编码 | 原型阶段：  分析、设计、编码 | 集成 | 测试 |

迭代+增量式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 了解部分需求 | 将收集到的需求原型化。收集反馈。选择技术架构  (持续迭代) | 完整实现3个功能，并随即集成 | 测试架构。演示现有功能 | 继续实现，并不断进行集成 | 最终集成 |

三种生命周期特点如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 生命周期类型 | 优势以及成功的必要条件 | 项目优先级 | 成功预期 |
| 增量式 | 管理日程风险  可以应对小的需求变更，但是对于会影响到架构的变更无法处理 | 1. 发布时间 2. 低缺陷率 3. 功能集合 | 成功 |
| 迭代式 | 管理技术风险  不断演化的需求 | 1. 功能集合 2. 低缺陷率 3. 发布时间 | 迭代中的任务已做规划，并且按计划完成 |
| 增量+迭代 | 管理日程和技术风险  如果团队成员不在同一物理位置，人员只能不完备，就难以实施 | 1. 发布时间 2. 功能集合 3. 低缺陷率 | 成功 |

## 常见问题解答

* 本指南是否适用于实训项目？

适用。教学部的学生实训课程与项目基地的实际项目采用统一的开发规范管理。而本指南是管理规范的纲领，所以即使是实训项目也应采取本指南的规范执行。

* 本指南是否适用于创新型项目？本指南是否会阻碍创新？

本指南适用于创新型项目。结合本指南的规范采用迭代式的生命周期，可以更有效地推进创新型项目的快速推出，缩短从理念到产品的距离，而不会阻碍创新。

* 本指南是否可裁剪？最小裁剪范围是什么？

可以，但应谨慎。本指南的内容已经在常规软件开发过程的基础上做了大量裁减，本指南只保留了能够有效推动快速交付有价值成果的最优时效的做法。如果需要剪裁的话，一定要遵守本指南的价值观，坚持Daily Build、持续集成和对外状态透明。

* 本指南是否可扩展？

可以。本指南中只包含适用于顺利完成大多数项目的最佳实践，力求实效。项目团队可以根据项目情况酌情扩展。

* 如何结合本指南与ICONIX软件过程？

ICONIX尤其是学院扩展的版本，是由需求挖掘到系统设计逐步推进的一个非常好的方法，但其对项目管理、质量控制、团队协作方面并未涉及，而本指南的重点就是在ICONIX未涉及的部分，因而ICONIX正好能够和本指南互相补充。并且本指南也推荐在每个迭代(增量/Sprint)中运用ICONIX方法。

* PMBOK的规范适合做为本指南的补充吗？

可以。本指南中一些内容也是根据PMBOK的知识编写的。项目团队也可以在讲PMBOK中的一些知识引入到项目管理过程中，尤其是关于时间管理、沟通管理、风险管理方面。但是请注意PMBOK只是一个知识体和指南，软件项目有其自身的特点，在引入PMBOK时必须结合项目特点和软件项目的特点。

## 参考文献

MSF [www.microsoft.com/msf](http://www.microsoft.com/msf)

项目管理修炼之道

Scrum [www.scrumcn.com](http://www.scrumcn.com)

PMBOK 2008版

ICONIX软件过程

## 修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 |
| 0.1 | 苑旭 | 2011-3-29 | 起草 |