目录

[一、 敏捷开发：9种主流开发方法介绍 1](#_Toc69655962)

[1、 极限编程 1](#_Toc69655963)

[1) XP的核心价值 1](#_Toc69655964)

[2) 为什么称为“Extreme”（极限） 1](#_Toc69655965)

[3) XP核心实践 1](#_Toc69655966)

[2、 水晶方法 2](#_Toc69655967)

[3、 动态系统开发方法 2](#_Toc69655968)

[1) DSDM的基本观点 2](#_Toc69655969)

[2) DSDM的基本原则 2](#_Toc69655970)

[4、 精益开发 3](#_Toc69655971)

[1) 精益开发的基本原则 3](#_Toc69655972)

[5、 Scrum 3](#_Toc69655973)

[1) Scrum 过程框架 3](#_Toc69655974)

[2) Scrum的四大支柱 4](#_Toc69655975)

[3) Scrum团队介绍 4](#_Toc69655976)

[4) 什么是用户故事 5](#_Toc69655977)

[6、 FDD 5](#_Toc69655978)

[7、 ASD 5](#_Toc69655979)

[8、 DSDM 5](#_Toc69655980)

[9、 轻量型RUP 6](#_Toc69655981)

[二、 传统项目管理VS敏捷项目管理对比 6](#_Toc69655982)

[三、 风险管理 6](#_Toc69655983)

# 敏捷开发：9种主流开发方法介绍

## 极限编程

极限编程（ExtremeProgramming，简称XP）是由KentBeck在1996年提出的。极限编程是一个轻量级的、灵巧的软件开发方法；同时也是一个非常严谨和周密的方法。

XP是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。

### XP的核心价值

XP的核心价值观是沟通（Communication）、简单（Simplicity）、反馈（Feedback）、勇气（Courage）、谦逊（Modesty）。 　XP用“沟通、简单、反馈、勇气和谦逊”来减轻开发压力和包袱；无论是术语命名、专著叙述内容和方式、过程要求，都可以从中感受到轻松愉快和主动奋发的态度和气氛。这是一种帮助理解和更容易激发人的潜力的手段。XP用自己的实践，在一定范围内成功地打破了软件工程“必须重量”才能成功的传统观念。

XP精神可以启发我们如何学习和对待快速变化、多样的开发技术。成功学习XP的关键，是用“沟通、简单、反馈、勇气和谦逊”的态度来对待XP；轻松愉快地来感受XP的实践思想；自己认真实践后，通过对真实反馈的分析，来决定XP对自己的价值；有勇气接受它，或改进它。

### 为什么称为“Extreme”（极限）

“Extreme”（极限）是指，对比传统的项目开发方式，XP强调把它列出的每个方法和思想做到极限、做到最好；其它所不提倡的，XP则一概忽略（如开发前期的整体设计等）。一个严格实施XP的项目，其开发过程应该是平稳的、高效的和快速的，能够做到一周40小时工作制而不拖延项目进度。

### XP核心实践

基于敏捷的核心思想和价值目标，XP要求项目团队遵循13个核心实践

团队协作：通过客户、开发团队、项目经理三方共同参加的会议来确定开发计划。

规划策略： 计划是持续的、循序渐进的。每2周，开发人员就为下2周估算候选特性的成本，而客户则根据成本和商务价值来选择要实现的特性。

结对编程：系统的每一行代码都是两个人用一个键盘完成的。

测试驱动开发：先写测试，后写代码。

重构：不断优化系统设计，使之保持简单。

简单设计：为明确的功能进行最优的设计，不考虑未来可能需要的功能。

代码集体所有权：开发队伍中任何人可以修改任何其他人的代码，代码不属于某个个人。

持续集成：至少每天将整个系统集成一次，保持一个能运转的系统。

客户测试：客户自己也是软件开发队伍的重要一份子。

小版本发布：尽快发布，尽早发布。

每周40小时工作制：保证休息，保持体力。

编码标准：必须有统一的编码规范，确保代码的可读性。

系统隐喻：将整个系统联系在一起的全局视图；它是系统的未来影像，是它使得所有单独模块的位置和外观变得明显直观。如果模块的外观与整个隐喻不符，那么你就知道该模块是错误的。

## 水晶方法

水晶（Crystal）方法论由Alistair Cockburn在20世纪90年代末提出。他把开发看做是一系列的协作游戏，而写文档的目标是帮助团队在下一个游戏中取得胜利。水晶方法的工作产品包括用例、风险列表、迭代计划、核心领域模型，以及记录了一些选择结果的设计注释。水晶方法也为这些产品定义了相应的角色。值得注意的是这些文档没有模板，描述也不太规范，但目标清晰，能够满足下次游戏开始的条件。

对于水晶方法论，根据方法论的轻重可以分为透明水晶和橙色水晶等。透明水晶一般适用于轻量级的团队。不管是哪种水晶，都会对团队的角色、团队的工作项和产出、核心实践、支持过程等进行定义。

## 动态系统开发方法

动态系统开发方法（DSDM）倡导以业务为核心，快速而有效地进行系统开发。可以把DSDM看成一种控制框架，其重点在于快速交付并补充如何应用这些控制的指导原则。

DSDM是一整套的方法论，不仅仅包括软件开发内容和实践，也包括了组织结构、项目管理、估算、工具环境、测试、配置管理、风险管理、重用等各个方面的内容。

### DSDM的基本观点

DSDM认为任何事情都不可能一次性圆满完成，应该用20%的时间完成80%的有用功能，以适合商业目的为准。实施的思路是，在时间进度和可用资源预先固定的情况下，力争最大化地满足业务需求（传统方法一般是需求固定，时间和资源可变），交付所需要的系统。对于交付的系统，必须达到足够的稳定程度以在实际环境中运行；对于业务方面的某些紧急需求，也必须能够在短时间内得到满足，并在后续迭代阶段中对功能进行完善。

### DSDM的基本原则

活动用户必须参与。

必须授权DSDM团队进行决策。

注重频繁交付产品。

判断产品是否可接受的一个基本标准是符合业务目的。

对准确的业务解决方案需要采用循环和增量开发。

开发期间的所有更改都是可逆的。

基本要求是高层次的并区分优先级（以在低优先级的项目上获得一定的灵活性）。

在整个生命周期集成测试。

在所有参与者之间采用协作和合作方法

## 精益开发

精益（Lean）管理的思想起源于丰田公司，旨在创造价值的目标下，通过改良流程不断地消除浪费。这种方法现已被广泛用于生产制造管理，对于IT系统建设，精益开发的常用工具模型是价值流模型。

### 精益开发的基本原则

杜绝浪费：将所有的时间花在能够增加客户价值的事情上。

推迟决策：根据实际情况保持可选方案的开放性，但时间不能过长。

加强学习：使用科学的学习方法。

快速交付：当客户索取价值时应立即交付价值。

打造精品：使用恰当的方法确保质量。

授权团队：让创造增值的员工充分发挥自己的潜力。

优化整体：防止以损害整体为代价而优化部分的倾向。

## Scrum

Scrum 是一个用于开发和维护复杂产品的框架 ，是一个增量的、迭代的开发过程。在这个框架中，整个开发过程由若干个短的迭代周期组成，一个短的迭代周期称为一个Sprint，每个Sprint的建议长度是2到4周。

在Scrum中，使用产品Backlog来管理产品的需求，产品backlog是一个按照商业价值排序的需求列表，列表条目的体现形式通常为用户故事。Scrum团队总是先开发对客户具有较高价值的需求。在Sprint中，Scrum团队从产品Backlog中挑选最高优先级的需求进行开发。

挑选的需求在Sprint计划会议上经过讨论、分析和估算得到相应的任务列表，我们称它为Sprint backlog。在每个迭代结束时，Scrum团队将递交潜在可交付的产品增量。 Scrum起源于软件开发项目，但它适用于任何复杂的或是创新性的项目。

### Scrum 过程框架

Scrum以经验性过程控制理论（经验主义）做为理论基础的过程。经验主义主张知识源于经验, 以及基于已知的东西做决定。Scrum 采用迭代、增量的方法来优化可预见性并控制风险。Scrum过程框架的基石包括如下三个方面：

第一：透明性（Transparency）。透明度是指，在软件开发过程的各个环节保持高度的可见性，影响交付成果的各个方面对于参与交付的所有人、管理生产结果的人保持透明。管理生产成果的人不仅要能够看到过程的这些方面，而且必须理解他们看到的内容。也就是说，当某个人在检验一个过程，并确信某一个任务已经完成时，这个完成必须等同于他们对完成的定义。

第二：检验（Inspection）。开发过程中的各方面必须做到足够频繁地检验，确保能够及时发现过程中的重大偏差。在确定检验频率时，需要考虑到检验会引起所有过程发生变化。当规定的检验频率超出了过程检验所能容许的程度，那么就会出现问题。幸运的是，软件开发并不会出现这种情况。另一个因素就是检验工作成果人员的技能水平和积极性。

第三：适应（Adaptation）。如果检验人员检验的时候发现过程中的一个或多个方面不满足验收标准，并且最终产品是不合格的，那么便需要对过程或是材料进行调整。调整工作必须尽快实施，以减少进一步的偏差。

Scrum中通过三个活动进行检验和适应：每日例会检验Sprint目标的进展，做出调整，从而优化次日的工作价值；Sprint评审和计划会议检验发布目标的进展，做出调整，从而优化下一个Sprint的工作价值；Sprint回顾会议是用来回顾已经完成的Sprint，并且确定做出什么样的改善可以使接下来的Sprint更加高效、更加令人满意，并且工作更快乐。

### Scrum的四大支柱

第一、迭代开发。在Scrum的开发模式下，我们将开发周期分成多个1-4周的迭代，每个迭代都交付一些增量的可工作的功能。迭代的长度是固定的，如果我们选择了1周的迭代，那么保持它的长度不要发生变化，在整个产品开发周期内每个迭代都是1周的长度。这里需要强调的是在每个迭代必须产出可工作的增量功能，而不是第一个迭代做需求、第二个迭代做设计、第三个迭代做代码。

第二、增量交付。增量是一个 Sprint 及以前所有 Sprint 中完成的所有产品代办事项列表条目的总和。 在 Sprint 的结尾,新的增量必须“完成”,这意味着它必须可用并且达到了 Scrum 团队 “完成”的定义的标准。无论产品负责人是否决定真正发布它,增量必须可用。增量是从用户的角度来描述的，它意味着从用户的角度可工作。

第三、自组织团队。Scrum团队是一个自组织的团队，传统的命令与控制式的团队只有执行任务的权利，而自组织团队有权进行设计、计划和执行任务，自组织团队还需要自己监督和管理他们的工程过程和进度，自组织团队自己决定团队内如何开展工作，决定谁来做什么，即分工协作的方式。

第四、高优先级的需求驱动。在Scrum中，我们使用Product Backlog来管理需求，Product Backlog是一个需求的清单，Product Backlog中的需求是渐进明细的，Backlog当中的条目必须按照商业价值的高低排序。Scrum团队在开发需求的时候，从Backlog最上层的高优先级的需求开始开发。在Scrum中，只要有足够1-2个Sprint开发的细化了的高优先级的需求，我们就可以启动Sprint了，而不必等到所有的需求都细化之后。我们可以在开发期间通过Backlog的梳理来逐步的细化需求。

### Scrum团队介绍

在Scrum的工作方式下，总共只有三个角色， 这三个角色分别是产品负责人（PO）,Scrum Master和开发团队。

我们通常可以以划龙舟的团队角色来类比Scrum的角色，划龙舟通常有舵手、鼓手、划桨团队三个角色。Scrum中的PO就是舵手的角色，他对产品的方向负责，对产品的Why和What负责，对产品的愿景，产品包括哪些主要的特性负责。Scrum中的Scrum Master鼓手的角色，他帮助团队保持高昂的士气，并进行良好的协作，他是一个Scrum的专家，团队的教练，团队的服务式领导。Scrum中的团队，对应到龙舟赛的划桨团队，团队必须协调一致，作为一个整体前进，在这样的环境下单打独斗，各自为政没有任何胜算。

Scrum的开发团队对实现Sprint目标需要做的所有事情负责，包括技术方案和决策，团队分工（谁做什么），执行Sprint开发任务等，而且作为自组织的团队，他们也对他们的工作进度的跟踪和管理负责。Scrum开发团队的主要职责包括如下五个方面：

执行Sprint

每日检视和调整，每个开发团队成员都应该参与每日站会，一起检验Sprint目标的进展情况，跟进当天的工作情况调整计划。

梳理产品列表，每个Sprint都需要花一些时间来准备下一个Sprint，主要用来梳理产品列表，包括PBI的创建和细化、估算和排列优先级顺序。

sprint规划，在Sprint计划会议（Sprint Planning Meeting）上，在ScrumMaster的引导下，开发团队和PO合作，为下一个Sprint建立目标。

检视和调整产品与过程，每个Sprint结束后，开发团队都要参加两个检视和调整的活动，即Sprint评审会议（Sprint Review Meeting）和 Sprint回顾会议（Sprint Retrospective Meeting）。

### 什么是用户故事

用户故事是从用户的角度来描述用户渴望得到的功能。一个好的用户故事包括三个要素：

角色：谁要使用这个功能。

活动：需要完成什么样的功能。

商业价值：为什么需要这个功能，这个功能带来什么样的价值。

用户故事通常按照如下的格式来表达：作为一个<角色>, 我想要<活动>, 以便于<商业价值>

## FDD

FDD（Feature-Driven Development，特性驱动开发）由Peter Coad、Jeff de Luca 、Eric Lefebvre共同开发，是一套针对中小型软件开发项目的开发模式。此外，FDD是一个模型驱动的快速迭代开发过程，它强调的是简化、实用、易于被开发团队接受，适用于需求经常变动的项目。

## ASD

ASD（Adaptive Software Development，自适应软件开发）由Jim Highsmith在1999年正式提出。ASD强调开发方法的适应性（Adaptive），这一思想来源于复杂系统的混沌理论。ASD不象其他方法那样有很多具体的实践做法，它更侧重为ASD的重要性提供最根本的基础，并从更高的组织和管理层次来阐述开发方法为什么要具备适应性。

## DSDM

DSDM（动态系统开发方法）是众多敏捷开发方法中的一种，它倡导以业务为核心，快速而有效地进行系统开发。实践证明DSDM是成功的敏捷开发方法之一。在英国，由于其在各种规模的软件组织中的成功，它已成为应用最为广泛的快速应用开发方法。

DSDM不但遵循了敏捷方法的原理，而且也适合那些成熟的传统开发方法有坚实基础的软件组织。

## 轻量型RUP

RUP其实是个过程的框架，它可以包容许多不同类型的过程，Craig Larman 极力主张以敏捷型方式来使用RUP。他的观点是:目前如此众多的努力以推进敏捷型方法，只不过是在接受能被视为RUP 的主流OO开发方法而已。

# 传统项目管理VS敏捷项目管理对比

传统项目管理通常采用的是瀑布式、部分迭代开发模式，要求在项目建设时，需求足够明确、文档足够规范，迭代过程中需求变更越多、越晚，对项目影响越大，会影响到项目的交付质量。

敏捷项目管理作为新兴的项目管理模式，简化了传统项目管理的繁琐流程和文档。以 Scrum 为代表，欢迎需求变更，在客户需求不明确的时候，以在较短的周期内开发出可用的软件为目标，来帮助客户描述自己的需求。迭代过程中的需求变更会加入到项目继续迭代需求池，丰富项目的产品功能。

敏捷项目管理VS传统项目管理的相同点：

* 敏捷项目管理声称要摆脱繁冗的流程制度文档，但是对于关键的项目文档，比如需求规格说明书等等，也是要求必须具备的。所以，敏捷项目管理的项目流程制度上的管理可以看作是对一套完善的项目管理流程制度的裁剪，只不过这个裁剪的尺度比较大，从而也对敏捷项目团队成员的适应性，自主性提出了较高的要求。
* 具体的敏捷方法在每个迭代周期中都存在立会制度，燃尽图、看板监控、计划发布等，这些和PMBOK中对项目生命周期的五个过程组启动、规划、执行、监控和收尾的定义没有冲突矛盾，实际上敏捷项目管理的这些措施可以看作是PMBOK项目生命期五个过程组执行的微缩版，区别在于敏捷项目管理的迭代周期，时间很短，在去执行过程中裁剪了很多规范正式的项目管理流程制度。

敏捷项目管理传统项目管理的区别

* 不同的管理方式适用于不同类型的项目，Scrum更适用于未知、不可知或持续变化的项目；
* 传统的管理方式有如计划经济体制，Scrum有如市场经济体制，适应变化的能力不同；
* 极大地缩短了用户与开发者，预期目标与实施状况，投资与投资回报之间的反馈回路；
* 将小型团队转化为自身命运的管理者，团队接受挑战，找寻应对挑战的方法，发挥创意，避开工作障碍，而这一切都无法由中央控制及调度系统预先安排。

# 风险管理

风险识别方法很多，常见的方法有：

* 生产流程分析法

生产流程分析法是对企业整个生产经营过程进行全面分析，对其中各个环节逐项分析可能遭遇的风险，找出各种潜在的风险因素。生产流程分析法可分为风险列举法和流程图法。

风险列举法指风险管理部门根据该企业的生产流程，列举出各个生产环节的所有风险。

流程图法指企业风险管理部门将整个企业生产过程一切环节系统化、顺序化，制成流程图，从而便于发现企业面临的风险。

* 财务表格分析法

财务表格分析法是通过对企业的资产负债表、损益表、营业报告书及其他有关资料进行分析，从而识别和发现企业现有的财产、责任等面临的风险。

* 保险调查法

采用保险调查法进行风险识别可以利用两种形式：

通过保险险种一览表，企业可以根据保险公司或者专门保险刊物的保险险种一览表，选择适合该企业需要的险种。这种方法仅仅对可保风险进行识别，对不可保风险则无能为力。

委托保险人或者保险咨询服务机构对该企业的风险管理进行调查设计，找出各种财产和责任存在的风险。