

An aerial photograph of a city with a grid-like street pattern, overlaid with a semi-transparent green filter. In the upper center, a group of skydivers is in formation, appearing as dark silhouettes against the lighter background of the city below.

Scrum 与 CMMI® 指南： 使用 CMMI 提高敏捷性能



CMMI® Institute

目录

引言	5
背景	8
什么是 CMMI?	12
了解 CMMI	14
什么是敏捷?	15
敏捷和 CMMI 如何协同作用以帮助产出更优秀的软件?	17
成功的敏捷过程改进项目的要素	23
使用 CMMI 共性实践让敏捷制度化	26
集成 CMMI 与敏捷实践和技术	28
待办事项梳理	31
持续构建 / 持续集成	32
每日站会 / 每日 Scrum	33
完成的标准	34
史诗	35
团队估算游戏（斐波那契游戏）与策划扑克	36
结对编程	37
产品待办事项列表	38
重构	39

发布燃尽图	40
发布策划.....	41
冲刺 / 迭代	42
冲刺燃尽图	43
冲刺演示 / 冲刺评审	44
冲刺策划.....	45
团队协议.....	46
技术债务.....	47
测试驱动开发	48
用户故事.....	49
速率	50
使用 CMMI 来增强敏捷成果	51
待办事项梳理	52
持续构建 / 集成	58
每日站会 / 每日 Scrum.....	62
完成的标准	67
史诗	69
团队估算游戏与策划扑克.....	73
结对编程	76
产品待办事项列表	81
重构	85

发布燃尽图	87
发布策划	91
冲刺 / 迭代	96
冲刺燃尽图	98
冲刺演示 / 冲刺评审	101
冲刺策划	107
团队协议	113
技术债务	116
测试驱动开发	118
用户故事	123
速率	126
参考资料、白皮书、演示文稿	129
开发团队	130

引言

越来越多的公司转用 CMMI 来提高敏捷举措的性能

能力成熟度模型集成 (CMMI®) 在全球的应用与日剧增，人们利用它来打造扩展性好、适应性强、性能高的组织，并促进组织兑现敏捷方法的承诺。很多组织利用 CMMI 来扩展和加强敏捷举措的实施并解决敏捷方法未涵盖的业务问题。全球各地的很多组织发现 CMMI 是实现敏捷性能的有效平台。CMMI 与生命周期无关，适用于任何软件开发方法。随着 CMMI 的应用日益广泛，在政府和商业领域的采用率与日俱增，它现已成为全球软件和系统工程组织的性能改进模型。

成功的组织同时需要敏捷性和稳定性

在 *Agility: It Rhymes with Stability*¹ 一文中，Wouter Aghina、Aaron De Smet 和 Kirsten Weerda 论述了真正的敏捷组织必须基于稳定的过程骨架，管理方法和组织结构建立动态的能力。若要同时兼备所需的稳定性和响应能力，组织必须“采用相对固定的核心元素设计组织结构、管理各种分工和过程 — 固定的主干。与此同时，他们必须创建可以快速适应新的挑战与机遇的较为宽松、更加动态的元素。”

组织乐于采用 CMMI 和敏捷方法来实现此看似自相矛盾的组合，以建立真正的组织敏捷性。通过采用 CMMI 所获得的纪律性、组织级知识和一致性，可以支持组织打造更加强大有效的敏捷方法。

CMMI 提供了高性能组织必须“做什么”的地图，敏捷方法则描述了“如何做”。随着方法和技术的适应与演变，CMMI 为组织奠定了基础，使组织可以据此采用适合组织业务环境动态变化的方式，重复利用技术或量身定制技术。对于软件工程师，就类似于将 CMMI 视为其组织的“需求”或“故事点”，而将各种敏捷实践或技术视为那些要求的具体实例化。

¹ <http://www.mckinsey.com/business-functions/organization/our-insights/agility-it-rhymes-with-stability>

组织利用 CMMI 提高敏捷性能

由于经过实践检验，CMMI 可带来令人称道的成果，越来越多在各种性能问题泥潭中挣扎的敏捷组织转而采用 CMMI。CMMI 提供了一个最佳实践模型，放眼未来审视团队性能，从而在系统层级应用精益原则。例如，Minacs IT Services 将 CMMI 应用到现有敏捷过程之后，冲刺承诺兑现率提高了 30% 到 40%，每次冲刺交付的用户故事数量增加了 30%，准时交付率提高了 40%。Minacs 还将内部工作文化从浓厚的单打独斗氛围扭转为拥有共同愿景、团结一心、目标一致的氛围。

组织使用 CMMI 识别过程与运营中的性能差距，并提供基于行业最佳实践进行持续改进的基线。通过弥合这些差距，组织使用 CMMI 建立稳定性，从而在项目和计划方面实现更高的敏捷性并降低成本、提高质量和准时交付率。

CMMI 帮助在整个组织内扩展和保持敏捷性

组织以 CMMI 为平台，扩展、调整 and 统一大型跨国公司分布在不同地理位置的运营机构之间的运营。例如，Cognizant 维持了 CMMI 成熟度 5 级评级，并使用 CMMI 和敏捷方法，在分布全球的整个组织中促进过程改进，从而满足以客户为中心的业务目标。

通过采用 CMMI 所获得的纪律性、组织级知识和一致性，让组织能够使用 CMMI 打造更加强大有效的敏捷方法。实际上，拥有 7,000 名工程师的霍尼韦尔印度公司将 CMMI 和敏捷方法运用于整个企业，从而提高问题解决技能并使问题止步于开发过程早期。所取得成果包括：功能性缺陷率降低 12-15%，改善策略的实现率提高 15%，员工学习曲线缩短。

使用敏捷方法的公司中 CMMI 的全球采用率迅速增加

仅 2015 年，CMMI 的全球采用率就增长了 17%，美国的增长率为 28%。在这一年，1,900 多家高性能组织获得了 CMMI 成熟度等级评级。作为提高组织性能的平台，CMMI 已在 100 多个国家内得到采用并获得 41 分的世界一流净推荐值，部署 CMMI 的公司对所取得的成果十分满意。

实施敏捷方法的组织中，CMMI 采用率在稳步增长。2015 年，大约 70% 的被评估组织宣称使用了一种或多种敏捷实践。[来源于 CMMI 研究所评估记录]。在美国、中国、印度和拉丁美洲设有技术中心的多家跨国公司在 使用 CMMI 扩展敏捷实践，并在全球范围内统一了他们分布在不同地理位置

运营机构之间的性能平台。CMMI 和敏捷方法在 Perficient Chennai 得到和谐应用。该公司使用 CMMI 将敏捷方法扩展到其北美、印度和中国的分公司，从而成功地将项目缺陷率降低 70%。将近 85% 的组织项目团队采用 CMMI 成熟度等级为 4 和 5 的实践以及敏捷方法，预测项目的性能和速度。

适用于政府以及大小私营公司的平台

尽管 CMMI 继续在航空和国防行业强势扩张（例如 GE Aviation、Boeing、Lockheed、Northrup Grumman、BAE Systems 和 Raytheon 等公司），但实际上采用 CMMI 的组织 90% 都在商业领域，包括移动、金融、电信和 IT 服务（例如 Honeywell、Samsung、Ericsson 和 Fujitsu 等公司）。

虽然 CMMI 很大程度上依靠大型跨国机构，但在高性能的小型业务单位中的采用率最高。实际上，68% 实施 CMMI 的组织的员工人数不到 100 名，22% 通过 CMMI 评估的组织的员工人数在 25 名以下。

CMMI 研究所开展研究来提高全球组织的性能

2012 年，在全球商业和政府采用率经过数十年持续增长之后，CMMI 研究所从卡内基梅隆大学的软件工程研究院分拆出来。2016 年，该研究所成为 ISACA 大家庭的一个组成部分，ISACA 是致力于 IT 治理、风险和网络安全的全全球非营利专业协会。这次结构性变革使得 CMMI 研究所有能力执行其宏大愿景：推进运营最佳实践方面的研究并提升全球范围内组织的性能。

这次转变扩大了 CMMI 研究所的行业和全球视角，对其研究、模型开发和战略方向做出了极大贡献。研究所与全球知名组织积极合作来推进实践状态，并兑现敏捷宣言承诺，从而培养真正的动态自适应高性能组织。

请访问 <http://cmmiinstitute.com/cmml-and-agile> 了解详情

背景



新世纪初发生的两个相互独立的事件，彻底改变了软件和系统工程的面貌。

第一个事件是发布卡内基梅隆大学在软件 CMM 方面早期工作的成熟成果——能力成熟度模型集成 (CMMI)。它引入更加广泛、更加全面的模型，采取以过程为中心的方法，使运营机构与运营目标一致，提高了组织性能和产品质量。

CMMI 的初始版本面向多个工程学科，但它在软件方面的根基是不言而喻的，并且在产品的早期版本中占据主导地位。

今天，我们拥有 CMMI 研究所的面向开发的 CMMI、面向服务的 CMMI、面向采购的 CMMI、人力 CMM 以及数据管理成熟度模型。

它们共同构成了一个平台，来提升整个价值链上的企业性能并密切配合软件与系统工程社区的需求。

第二个事件是发布敏捷宣言。与 CMMI 的基于实验研究的方法相比，《敏捷宣言》的缔约者倾向于通过协作找到共同点，同时信奉以人为本的核心价值观，以支持开发相互独立的新技术和方法。

创始小组内的倡导者提出了很多敏捷开发方法，但如今最常被采用的方法是 Scrum 和极限编程 (XP)。

早期采用者

在早期，采用者的文化背景不同，造成美国国内对 CMMI 和敏捷之间的关系理解混乱。CMMI 与方法无关，无论是使用传统的工

作管理方法（例如“瀑布式”），还是诸如 Scrum、看板、关键链或螺旋之类的其他方法，都可用于提高组织性能。

在全球普及以及被业内很多大大小小的组织用作平台之前，CMMI 一开始被美国国防和航空行业使用。后来，商业领域的很多大型组织亦纷纷采用，包括自动化行业和含嵌入式系统的产品制造商。这些早期采用者运用 CMMI 来改善原有的传统的“瀑布式”运作模式以取得显著成效。这引起了一些混淆，甚至是在使用 CMMI 规范现有实践的组织内。一般人开始会将 CMMI 误解为其他方法的替代方法，而不是一种灵活的模型，无论组织采用何种方法，它都可将方法与组织目标保持一致、弥合价值链中的差距，并建立持续改进的文化来提升组织的战略一致性和性能。

“这些早期采用者会选择适当运用 CMMI — 用于改进它们现有业务开展方式的一种模型。”

在软件开发的其它领域，敏捷运动开始蓬勃发展。被视为改善协作与充分利用个体优势以及个人责任（“高信任”）的一种途径，敏捷采用率在过去十年间呈指数增长，大部分用于纠正当时大多数软件开发组织所偏爱之传统方法的“低信任”特点。

敏捷方法规避了“瀑布式”组织喜爱的许多传统事件和行为，选择放弃项目审计和传统衡量标准，例如，利用结构化行为代替小型团队中员工的生产效率和高技能团队协作。

目前，最受欢迎的敏捷方法是 Scrum，这种协作方法用于管理强调固定时间箱短期迭代（“冲刺”）工作、所分配的小型、稳定团队、短期策划以及严格的对等责任。

因此，我们所得出的结论是 Scrum 的早期采用者当中有很多是小型组织或大型组织中的小团队。极限编程（“XP”）是另外一种应用广泛的敏捷方法，它起源于一家公司内部，就是后来成为 CMMI 最大采用者之一的克莱斯勒汽车公司。

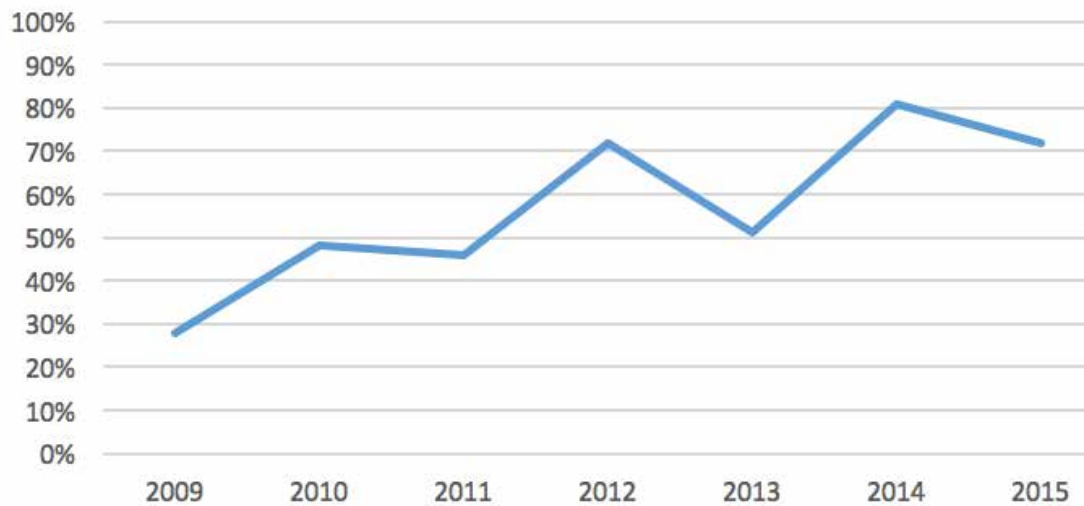
幸运的是，随着敏捷方法的应用日益广泛，不受 CMMI 和敏捷历史遗留文化预设所羁绊的组织，开始使用 CMMI 来克服敏捷实施中扩展性、一致性、透明性、适应性和性能方面的挑战。而已经使用 CMMI 的组织则用它来井然有序地实现敏捷转型，实现它们所追求的稳定性和动态能力。

本文档旨在向从业人员阐明他们可以如何利用 CMMI 平台来兑现对于敏捷价值观的承诺。

“大型早期采用者推动市场认识和 CMMI 认识，而敏捷一直由二者推动。”

“部分对于 CMMI 的认识源于早期采用者庞大的规模和业务范围。”

包含敏捷成分的 CMMI 评估百分比



资料来源：CMMI 研究所人员执行的 2015 研究项目在 PARS 中分析了 2015 年以来的一千多个 SCAMPI 评估，参考了 Scrum、极限编程、敏捷以及在基本单元和支持功能组的描述中的其他关键字的使用。

Scrum 与 CMMI 指南： 使用 CMMI 提高敏捷性能

后续文档是在敏捷环境中应用 CMMI 的路线图。虽然敏捷框架可以成功用于管理几乎任何类型的工作，但它们在软件和系统开发行业的应用最为广泛，因此本文档采用 CMMI-DEV、Scrum 和极限编程作为基准。

受众

本文档旨在帮助三类互不相同的群体应用 CMMI 来提高性能。它们分别由以下用户故事代表：

作为一家敏捷组织

我们希望了解如何使用 CMMI 来提高性能

以便我们可以推动敏捷项目

作为一家敏捷组织

我们希望了解采用 CMMI 的最佳方式

以便在提高的同时取得 CMMI 评级

作为 CMMI 主任评估师

我希望了解如何在敏捷环境中应用 CMMI

以使我更有效地执行敏捷组织的评估

为了满足这三类需求，本文档编排了以下部分：

- I. 使用 CMMI 时调整并强化敏捷实践和技术
- II. 使用 CMMI 时调整并强化敏捷价值观
- III. 敏捷无法解决的 CMMI 中的性能改进实践
- IV. 参考资料、白皮书、案例研究和示例

我们的建议是您通过选择特定的实践或技术，跳转到相关部分，然后运用那里提供的一些实践或建议，从而逐步了解 CMMI 如何强化敏捷。将此参考资料用作指南，而不是规则手册。考虑每项建议的意义，并自行决定它们可能为您、您的团队以及您的整个组织带来多大价值。根据您尝试实现的价值校正改进强度，然后继续下一部分。不要考虑成熟度级别，只需反复

并逐渐深入了解 CMMI 如何帮您提高团队性能、减少混乱并打造优秀的产品。

不要“只是做 CMMI”。着眼于敏捷，效果更佳。

什么是 CMMI ?

能力成熟度模型集成亦称 CMMI，是一个过程模型，明确定义了一个组织应该采取什么行动来定义、理解和推进有助于提高性能的行为。

面向开发的 CMMI 有五个“成熟度级别”和三个“能力级别”，定义了打造伟大产品和服务所需展现的最重要实践，并将它们全部囊括在一个综合模型中。

CMMI 还帮助公司确定并实现可度量的业务目标，打造更出色的产品，保证客户更满意并确保我们尽可能高效地工作。能力成熟度模型集成 (CMMI®) 是一种能力改进模型，为组织提升性能提供指南。CMMI 有 5 个成熟度级别，为了持续改进每个级别都建立在前一个级别的基础上。基于 300 多条实践可以对组织能力或成熟度级别进行“评级”。朝着更高成熟度级别努力的组织可以提高能力并推行更有效的过程。CMMI 的力量在于帮助组织识别差距，改进这些差距的能力可以产生巨大的影响。使用 CMMI 来培养能力的组织提高了质量，提升了客户满意度，留住了更多员工以及提高了盈利能力。

“ CMMI 是强化敏捷的绝佳选择，
可以扩展任何规模或复杂程度的
项目并促成其成功 ”



了解 CMMI

面向开发的 CMMI

面向开发的 CMMI 是一个参考模型，涵盖产品与服务开发活动。来自包括航空、金融、计算机硬件、软件、国防、自动化制造以及电信等众多行业的组织在使用面向开发的 CMMI。

面向开发的 CMMI 包含的实践涵盖项目管理、过程管理、系统工程、硬件工程、软件工程和开发及维护过程中使用的其他支持过程。

使用专业的判断和常识解读适合您的组织的模型。换句话说，虽然此模型所述的过程域反映的行为是为大多数用户考虑的最佳实践，但应根据对 CMMI-DEV、您的组织的局限以及您的业务环境的深入了解，解读过程域和实践。

过程域

过程域是一个领域中相关实践的集合，一起实施时，可实现对改进该领域十分重要的一组目标。

共性目标

共性目标之所以被称为“共性”，是因为相同的目标陈述适用于多个过程域。共性目标描述必须要具备哪些特征，才能制度化实施某个过程域的过程。共性目标是必备模型组件，用于在评估中判定过程域是否已被满足。

例如，共性目标可以是“将过程制度化为已定义过程”。

共性实践

共性实践之所以被称为“共性”，是因为相同的实践适用于多个过程域。与共性目标关联的共性实践描述在实现共性目标的过程中被视为十分重要的活动，并有助于制度化与过程域关联的过程。共性实践是一个期望模型组件。

什么是敏捷?

敏捷是一套价值观和一组相关的用以管理和交付项目的技术与方法的集合。敏捷强调迭代和增量开发、与客户密切合作以及跨职能团队，以构建可工作的软件。根据《敏捷宣言》中阐述的一套价值观和原则，敏捷开发包括 Scrum 和极限编程 (XP) 之类的一些框架。

敏捷运动不断演变以应对采用更加传统的瀑布式方法（常用于大型软件开发项目）的组织所遭遇的一些挫败。很多人认为瀑布式过程文档任务繁重并且成本高昂。瀑布式开发基于大量的前期策划并且在按步骤顺序完成项目的每个阶段后，方能前进至下一个阶段。瀑布式开发中使用的线性方法，与敏捷开发中使用的增量式和迭代式经验方法形成鲜明对比。

敏捷价值观

敏捷演变出一套与瀑布式方法截然不同的理念。《敏捷宣言》陈述了构成敏捷的核心价值，并强调以下几点：

- 人和互动胜过过程和工具
- 可以工作的软件胜过完备的文档
- 客户合作胜过合同谈判
- 应对变化胜过遵循计划



敏捷工作还遵循以下原则，这在《敏捷宣言》中亦有阐述：

- 通过快速交付可用的软件赢得客户满意
- 欢迎改变的需求，即使是在开发后期
- 频繁交付可工作的软件（按周而不是按月）
- 业务人员和开发人员之间每天密切合作
- 项目围绕积极向上的人员构建，并且应信任他们
- 面对面谈话是最好的沟通方式（身处同一地点）
- 可工作的软件是进度的主要度量指标
- 可持续开发，能够保持不变的步调
- 持续追求技术卓越和优异的设计
- 简单 — 最大化不必完成的工作量的艺术 — 是基本原则
- 自组织团队
- 定期调整以适应不断变化的环境

这些价值观和原则指导敏捷运动以及所涉及的各种方法和工具，包括 Scrum、极限编程、水晶方法以及动态系统开发方法 (DSDM)。

敏捷问题

凭借合作与迭代方法，敏捷价值观、实践和技术得到广泛认可。但也涌现一些问题。对于寻求采用敏捷方法的组织，往往很难量化了解实施敏捷方法的真实价值和投资回报。尽管增量式和迭代方法具有很高的价值，但是成功的标准通常是传闻性的和定性的。这对很多组织的管理方法与结构带来挑战，并推动了与敏捷价值观相悖的行为，因为管理层让项目组背上了使用传统度量元的包袱。

此外，大型采用者可能提出与敏捷价值观、实践和技术冲突的需求，推动威胁敏捷方法的变革。致力于践行敏捷价值观的组织正在想方设法为其敏捷实施灌输更强适应性，以抵消新的大型采用者带来的负面影响。

敏捷和 CMMI 如何协同作用以帮助产出更优秀的软件？

下表列出敏捷实践 / 技术和 CMMI 提供改进指南的一些常见业务问题。确定您需要解决的问题，进而选择您需要用来解决该问题的敏捷实践 / 技术。使用 CMMI 过程域强化敏捷实践 / 技术以确保可扩展性和适用性。

问题	敏捷实践 / 技术		CMMI 过程域
项目估算值不切实际或未知。	- 团队估算游戏 - 策划扑克 - 冲刺策划 - 待办事项梳理	- 需求开发 - 任务估算 - 发布策划 - 冲刺待办事项	- 项目策划 - 集成项目管理 - 需求管理和开发 - 度量分析
提示： 开始工作之前，应该使用团队估算游戏或策划扑克估算产品待办事项列表中的高优先级用户故事，以确定即将进行的冲刺要包含哪些用户故事；这将有助于了解需求的复杂程度和成本。 在发布和冲刺策划期间，如果团队大体没变，则应该考虑团队速度。这将让团队能够选择每个冲刺中要包含的用户故事点的合适数量。			
项目无法按时间表如期交付	- 每日站会 / 每日 Scrum - 发布燃尽图 - 冲刺燃尽图	- 任务估算 - 按需发布 - 增量发布	- 项目监督控制 - 度量分析
提示： 在项目期间，应该进行定期待办事项梳理以继续针对即将进行的冲刺分解用户故事。有些组织通过更为频繁的针对较小用户故事和任务的“微型修整”会议，取得了积极成果。 作为项目监控的一个部分，每日站会应该评审燃尽图，以了解发布 / 冲刺是否按计划进行。			

问题	敏捷实践 / 技术	CMMI 过程域
需求总是在变。	- 待办事项梳理 - 冲刺策划 - 产品待办事项列表 - 用户故事 / 史诗故事 - 完成的标准（针对用户故事）	- 需求管理 - 项目监督控制 - 需求开发 - 度量分析 - 验证
提示： 项目应该制定产品待办事项列表以跟踪所有用户故事（需求）。产品所有者（客户）负责保证此待办事项列表为最新，因此，如果需求有变，应该更新待办事项列表。在每个冲刺策划会议期间，都会选择要纳入的用户故事。冲刺策划完成后，应该冻结冲刺待办事项列表。冲刺 / 迭代期间提出的任何变更都应该归入产品待办事项列表。在项目期间，应该进行定期待办事项梳理以继续针对即将进行的冲刺分解用户故事。		
客户不会致力于项目。	- 发布策划 - 冲刺策划	- 需求管理 - 项目策划
提示： Scrum 产品所有者（SPO）要么是客户，要么是客户代表 / 代理。SPO 是发布策划和冲刺策划会议的主要参与者，这些会议上将选择要处理的用户故事。项目工作应在 SPO 同意所选择的用户故事后开始。所有用户故事都应规定验收标准。在每个冲刺 / 迭代期间，都设有冲刺演示 / 评审，将已完成的工作提交给客户进行验收。		
客户对进度感到失望。	- 发布策划 - 冲刺策划 - 冲刺演示 - 冲刺回顾	- 集成项目管理 - 确认 - 项目策划 - 项目监督控制 - 度量分析
提示： 顾客是项目的产品所有者时，他们将是团队的一份子，失望的可能性会降低。产品所有者应该定期参与发布策划、冲刺策划和冲刺演示 / 评审，以便他们始终掌握有关项目状态的最新信息并有足够的机会表达关切。对于每个项目故事，项目应该明确完成的标准，以确保验收标准和测试用例等事项完成情况一致。		

问题	敏捷实践 / 技术	CMMI 过程域
客户对最终产品不满意。	• 完成的标准 • 用户故事 / 史诗故事 • 测试驱动开发 • 冲刺演示	• 确认 • 验证 • 需求开发 • 需求管理
提示： 所有用户故事都应该制定可验证的验收标准。用户故事的确认应在冲刺演示 / 评审期间进行。如果这两项事情都无法让客户满意，则应该增加新的用户故事添加到产品待办事项列表，以便在将来予以解决。		
项目期间或项目之后有太多缺陷。	• 测试驱动开发 • 持续构建 / 持续集成 • 重构	• 确认 • 验证 • 产品集成 • 技术解决方案
提示： 持续构建 / 持续集成、测试驱动开发以及结对编程等技术都可以改善交付质量。工作开始之前，用户故事应该明确制定验收标准。这将确保建立相应的测试用例。 所有敏捷团队都应该制定“完成的标准”，其中包含如何推进用户故事以及何时可以结案的指南。这可能包括有关测试和缺陷的指南。 冲刺 / 迭代期间发现的缺陷可以通过多种方式消除。应该在每日站会中讨论这些缺陷，项目团队应该确定冲刺 / 迭代期间能否消除缺陷。如果故事带着缺陷结案，则可以在产品待办事项列表中创建新的用户故事以解决该已知缺陷。 重构也可用于改进产品，但仅限有意识地决定在冲刺策划期间提供较低品质的代码，并计划在稍后重新设计或重新编码以提高代码质量。策划项目时应考虑技术债务，以便团队留出时间修复缺陷和改进技术，从而满足产品的长远需求。		
无法保障和维持项目资源。	• 团队协议 • 发布策划 • 每日站会 • 愿景	• 集成项目管理 • 项目策划 • 风险管理 • 项目监督控制 • 组织级培训
提示： 发布策划期间，开展相关工作以确保您为项目提供足够的具备整套适当技能的团队成员。项目期间，如果存在资源问题阻碍进度，则应该在每日站会上提出。		

问题	敏捷实践 / 技术	CMMI 过程域
项目缺乏内部和外部支持。	- 团队协议 - 发布策划 - 冲刺策划 - 每日站会 - 产品所有者与客户的会议	- 集成项目管理 - 风险管理 - 项目策划 - 项目监督控制
提示： 项目团队应该参与发布和冲刺策划会议；这有助于确保每个人认同项目。项目团队可以创建团队协议，这有助于给团队行为提供指导。项目期间，应该在每日站会上提出所有疑虑。		
不良的项目沟通。	- 每日站会 - 发布策划 - 冲刺策划 - 冲刺演示 - 冲刺回顾 - 产品所有者与客户的会议	- 项目监督控制 - 项目策划 - 集成项目管理
提示： 采用敏捷方法的主要优势之一是其处理实时沟通的优越方式。发布 / 冲刺策划、每日站会、冲刺演示 / 评审和回顾等敏捷实践和技术鼓励团队成员每日进行增量式沟通。任何有关沟通频率的其他决定都可以在团队协议中找到。		
需求是模糊或开放性的。	- 用户故事 / 史诗故事 - 完成的标准 - 待办事项梳理	- 需求管理 - 需求开发
提示： 史诗故事通常被视为高级、复杂或描述不完整的一类用户故事。史诗故事应该在待办事项梳理或发布策划会议期间完善，直至信息详尽到足以进行估算的程度。应该对所有用户故事（需求）进行估算并包含“完成的标准”，然后才能在冲刺策划会议中进行选择。项目团队“完成的标准”应该包括验收用户故事的准则		

问题	敏捷实践 / 技术	CMMI 过程域
项目团队责任不明确。	- 团队协议 - 冲刺策划 - 发布策划 - 增量发布	- 集成项目策划 - 项目策划
提示： 项目团队责任可以包含在团队协议或完成的标准中。冲刺策划会议期间，所有用户故事都应该分解成多项任务，所有者自己领用任务。冲刺 / 迭代期间，每个团队成员负责报告自己分配的任务。		
项目团队成员针对任务的培训不足。	发布策划	- 组织级培训 - 项目策划
提示： 发布策划期间，团队成员开始了解发布的范围和内容时，应该在策划期间识别和考虑团队成员所需的培训。		
无法成功计划项目。	- 发布策划 - 冲刺策划 - 待办事项梳理	- 项目策划 - 集成项目策划 - 需求管理
提示： 发布策划和冲刺策划等 Scrum 实践（如果完整执行）确保有效进行项目策划。但执行实践还不够，应该制定并遵守与估算、团队性能历史、当前团队知识和团队行为有关的准则。		
无法预见潜在问题。	- 每日站会 - 发布策划 - 冲刺回顾	- 项目监督控制 - 风险管理 - 集成项目管理
提示： 每日站会（每日 scrum）是一项出色的工具，可让团队增量式地迭代确定风险和问题。Scrum 教练应该用心地从团队成员身上探明这些风险和问题，以防他们落入简单的“我做了什么、我在做什么、我有什么困难”的套路，但不提供重要内容。在每日站会期间提出并讨论风险和问题十分重要。任何符合技术债务条件的事项都应该作为风险加入产品待办事项列表，并记录为故事或一些其他工作单元。		

问题	敏捷实践 / 技术	CMMI 过程域
不能在需要时获得项目信息。	- 用户故事 / 史诗故事 - 冲刺策划 - 发布策划 - 每日站会	- 发布燃尽图 - 冲刺燃尽图 - 冲刺演示 - 愿景
		- 项目监控 - 集成项目管理 - 风险管理 - 度量分析 - 验证
提示： 实际上，每日站会应该每天进行，时长约为 15 分钟。核心 Scrum 团队成员、产品所有者和其他相关利益相关方应该按计划出席每日站会。每日站会不应该仅仅因为团队成员可能在某天的该时段无法参加而取消。应该充分了解、估算所有用户故事（包括验收标准），并能够在冲刺期间完成。项目团队“完成的标准”应该包括可接受用户故事规模的条件。冲刺策划期间，所有用户故事都应该细分为多项任务。一旦详细程度达到此级别，就可以使用燃尽图进行跟踪。这应该确保在适当的级别跟踪项目进度。		
代码质量不佳。	- 结对编程 - 测试驱动开发 - 持续构建 / 集成 - 回顾 - 完成的标准	- 技术解决方案 - 验证 - 确认 - 集成项目管理 - 过程和产品质量保证
提示： 有多种敏捷技术可以改进代码质量，包括结对编程（一人编码，另一人浏览并监督代码质量的结构）和持续构建 / 集成（其中的自动回归测试将产生经过改进的最终版本）。		
组织级性能未改进。	回顾	- 集成项目管理 - 组织过程焦点 - 组织过程定义 - 组织级培训
		- 过程和产品质量保证 - 组织过程性能 - 定量项目管理 - 因果分析和解决办法
提示： 所有敏捷方法的中心在于回顾的概念。“回顾”对于迭代和增量改进团队性能十分有用，但对于整个企业范围的变革影响甚微。		

成功的敏捷过程改进项目的要素

敏捷团队依靠“回顾”频繁提供增量信息，实现团队性能改进。虽然在“Scrum of Scrum”环境（多组 Scrum 团队同步工作解决一个团队中多个控制方面的问题）中部署一系列回顾在有些敏捷环境中已被证明行之有效，但敏捷中并没有机制来明确解决“企业级”改进问题。要在任何环境下（包括敏捷环境）以实用、有效的方式改进组织过程，一种经实践检验的方法是从多个不同组的参与者中抽调成员建立集成的过程团队（有时也称为“软件工程过程组”或“SEPG”），并由一个或多个工作组提供支持。这些工作组侧重于敏捷实践、技术或其他需要定义或改进性能的零散领域。

过程行动小组 (PAT)

侧重改进敏捷实践、技术或其他过程的工作组的通用术语是“过程行动小组”。其他常用术语包括“特殊兴趣小组”、“过程改进小组”或“敏捷行动小组”。PAT 是一个持久小组，它拥有、完善并制度化具体的敏捷实践、技术或其他过程。PAT 设计、开发并文档化子过程，召开并主持培训会议，采取排定好优先顺序的纠正措施并维护现有的过程资产。敏捷团队通过采用这项技术，顺利完成过程改进任务的分工实施。

每个 PAT 都由类似一个跨职能支持小组构成，有一个 PAT 组长（过程所有者）以及组织中该过程域的一名或多名主题专家 (SME)。每个 PAT 收集、分析和报告组织使用其过程域的相关度量数据并据此行动。

与过程改进项目的关系

组织过程定义 (OPD)

OPD 设定对于组织标准实践集架构的高层期望，而这些期望将作为 SEPG 使命的输入。因为在软件工程中，良好的过程架构可以为将来的成功奠定敦实基础。该架构应该足够灵活，让用户能够通过多种途径取得成果。“一刀切”的过程方法是解读 CMMI 时最常犯的错误，并且毫无疑问地会在敏捷环境中失败。极少组织有规模和范围完全一样的项目。不要犯这种错误！

组织过程焦点 (OPF)

OPF 设定对于改进项目本身架构的高层期望，并且这些期望将作为在业务、SEPG 和 PAT 之间划分执行过程创建 / 改进工作的责任的输入。

SEPG 与业务合作识别组织总体的过程需求和改进需求。SEPG 与 PAT 合作创建改进计划，然后予以执行。最后，在做项目的过程中，SEPG 和 PAT 实现改进，并确保它们对业务达成预期效果。目标应该是以短周期增量进行检查和调整。

集成项目管理 (IPM)

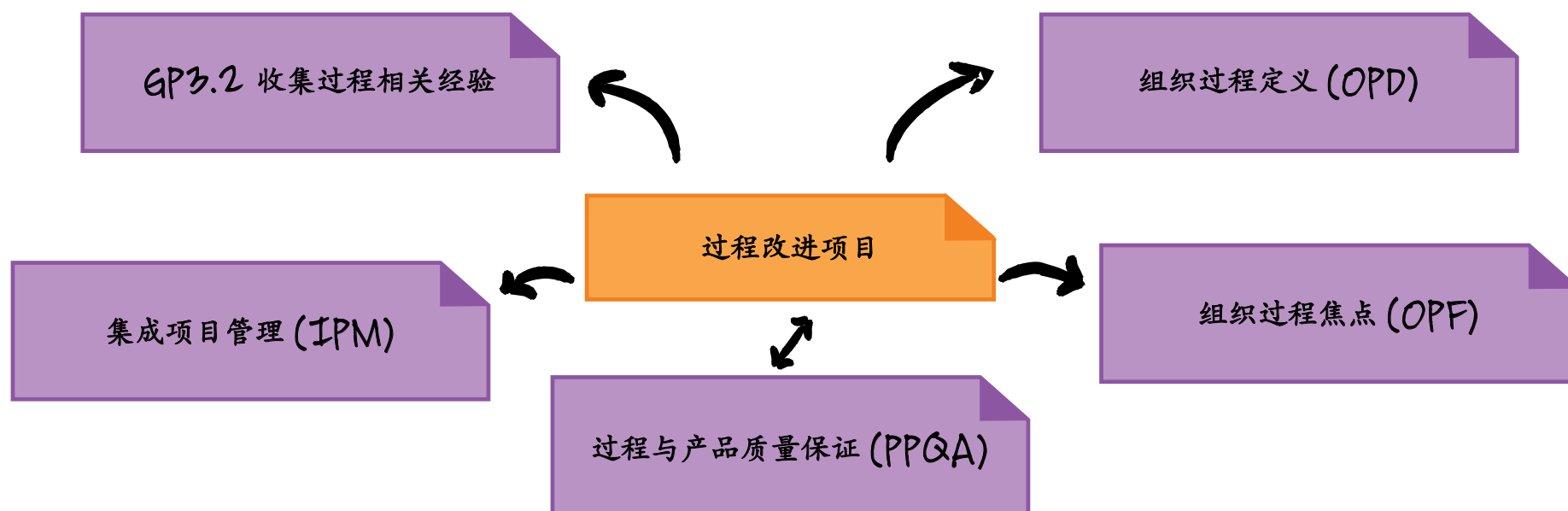
IPM 设定对于项目如何选择和吸收过程改进项目的输出成果的高层期望。做得好时，项目应

该能够选择最适合其独特环境的因素，同时具备有效和高效的结构。

收集过程相关经验

检查和调整的敏捷价值观是 GP3.2 “收集过程相关经验”的核心。支持此行为的过程改进项目会给项目和业务回报巨大价值。很多时候组织会收集经验教训，但之后不会明确相关的

责任人，又任其消失在黑洞中。在架构良好的过程改进项目中，SEPG 是过程相关经验的首选渠道。效果良好的新方法将被添加到相应的架构位置，需要改进的现有过程也可以作为任务抽离出来分配给责任 PAT。按这种方式，反馈环路是闭合的，经验教训会带来行为改变和业务改进。



敏捷环境中的过程改进

Scrum 等敏捷方法最适合运行过程改进项目。很多敏捷实践可轻松调整适应以支持过程改进项目。例如：

- 可以创建含用户故事的产品待办事项列表，从用户视角和组织视角定义过程需求。敏捷组织很有可能具备产品待办事项列表的工具和方法。
- Scrum 可用于计划和管理 SEPG 以及支持计划的任何小组的工作，比如过程行动小组。同样，敏捷组织很有可能具备 Scrum 的工具和方法。
- 策划扑克法可用于待办事项梳理和冲刺策划。
- 燃尽图和燃起图等标准信息辐射器可用于显示进度。
- 冲刺评审可用于向过程改进项目的客户展示新的过程和改进。
- 根据业务能够在改进方面投入的资源，可以增加冲刺持续时间，也可以将过程改进工作时长限定为每周最长几个小时。

任何敏捷过程改进项目的目标都应该是提高可预测性和透明化，制定关键执行主管信任并支持的改进计划。

使用 CMMI 共性实践让敏捷制度化

CMMI 的“共性实践”提供适用于协助组织制度化敏捷价值观、框架、实践和技术的指南和内容。 共性实践意在应用于敏捷实施的方方面面，被分为三种“共性目标”，每一种为组织成熟度提供相应级别的指南。

共性目标	共性实践	说明
实现特定目标 (GG1)		GG1 为刚刚开始敏捷或过程改进的组织提供指南，并指导团队开始侧重于实现每个 CMMI 过程域内的目标。
GG1	执行特定实践	GP1.1 指导刚刚开始敏捷或过程改进的组织，以确保他们在执行每个实践或技术内的所有适当实践。
制度化已管理的过程 (GG2)		GG2 帮助组织开始在多个项目或 scrum 团队中制度化敏捷目标。
GG2	制定组织级方针 (GP2.1)	GP2.1 协助组织设定对性能以及敏捷价值观（例如透明化、合作、快速失败、尽早失败、迭代式和增量式，以及人员胜过程和相关框架、实践和技术）采用情况的明确期望。
GG2	计划过程 (GP2.2)	GP2.2 通过明确阐述每个敏捷框架内的定义、各种技术与实践的使用和执行来协助敏捷团队。
GG2	提供资源 (GP2.3)	GP2.3 指导组织领导层和团队提供适当的资源以支持敏捷价值观。示例包括协作的工作空间、一致的团队、Scrum 教练和工具。
GG2	分配责任 (GP2.4)	GP2.4 帮助确保产品所有者、Scrum 教练和团队成员都清楚自己的责任，并被授予一定权限以有效履行他们的职责。
GG2	培训人员 (GP2.5)	GP2.5 指导敏捷组织为产品所有者、Scrum 教练和 Scrum 团队成员提供相应级别的培训，以便他们能够有效履行自己的职责。

共性目标	共性实践	说明
GG2	控制工作产品 (GP2.6)	GP2.6 对通过采用敏捷技术所获得的工作产品的配置控制提供指南, 协助敏捷团队。
GG2	识别利益相关方并使其参与进来 (GP2.7)	GP2.7 是敏捷团队的基础, 要求项目参与者稳定且始终如一地参与, 方能成功。参与者不限于 Scrum 团队, 包括冲刺演示和回顾在内的所有实践中的参与者都应考虑在内。
GG2	监督和控制过程 (GP2.8)	GP2.8 指导敏捷团队监控过程本身的性能。示例包括: 我们在履行冲刺承诺吗? 在按计划进行冲刺策划、回顾和冲刺演示吗? 我们引发了多少技术债务? 通过执行 GP2.8 收集的信息对于确保回顾成功至关重要。
GG2	客观评价符合性 (GP2.9)	GP2.9 对于新的敏捷团队而言是一个非常重要的实践, 这些团队有时需要协助澄清实施各种实践和技术的方法。GP2.9 为组织提供服务以确保团队的工作方式支持敏捷价值观。
GG2	与高层管理者评审状态 (P2.10)	管理层迫切希望了解成功执行敏捷举措所带来的成果, 特别是关于敏捷价值观的展示。GP2.10 提供与较高级的管理层分享信息的指南。
制度化已定义的过程 (GG3)		GG3 帮助组织继续成长, 从基础的由个别团队采用敏捷方法, 发展到整个企业范围的架构方法。
GG3	建立已定义的过程 (GP3.1)	GP3.1 为 Scrum 团队提供相关准则, 以基于组织标准和说明量身定制过程, 从而满足各自的特定目标。GP3.1 是扩展敏捷方法并在整个企业推行的关键。
GG3	收集过程相关经验 (GP3.2)	回顾是在项目层级获取经验和教训的主要机制。GP3.2 提供将该经验扩展到组织层级的准则, 以便所有团队都能从在团队层级所学的知识中受益。

集成 CMMI 与敏捷实践和技术

敏捷价值观

透明化。合作。人员胜过过程。快速失败。迭代式和增量式。《敏捷宣言》的作者付诸纸面的这些价值观，明确定义了对所有敏捷团队的期望。当这些价值观可追溯至敏捷工具箱中的相关框架和技术并与之融为一体时，即为团队提供了全功能过程框架。

这些价值观在 Scrum 和极限编程 (XP) 中尤为显著，其中的敏捷价值观与它们所支持的框架和技术之间存在直接可追溯性。

CMMI 主任评估师的观点

敏捷价值观展现在整个 CMMI 模型中，在 CMMI 模型中存在大量机会来强化和完善它们。在共性目标“制度化已管理的过程”(GG2)和“制度化已定义的过程”(GG3)内尤其如此，上述二者都对敏捷价值观的实施和采用以及直接追溯到它们的框架和技术提供指南。

敏捷价值观在 CMMI 术语中可被视为“方针”，假设相关框架和技术确实与它们一致。

传统的“瀑布式”组织可能通过公司方针设定期望，而敏捷组织可能通过声明 Scrum 为其标准并确保所有从业人员采用它以及相关敏捷价值观来设定。通过这种方式，敏捷价值观到敏捷技术的可追溯性就是“双向的”，也就是说采用任何一方亦会追溯到另一方。

敏捷价值观

实践 / 目标	强化敏捷实施的方式：	能够符合实践 / 目标，前提是：	体现方式：
制度化已管理的过程 (GG2) 共性实践： GP2.1 GP2.2 GP2.5 GP2.6 GP.2.7 GP2.9 GP2.10	通过明确定义组织所有层级可接受的价值观，让敏捷价值观的采用规范化。应该通过一项计划提供支持，以针对敏捷价值观的首秀、培训、识别相关利益相关方、在项目评估期间提供指导以及管理层评审项目性能。	价值观得到所有从业人员的充分理解，并在整个项目和支持小组内以一致的方式采用相关框架和技术。	培训材料、方针定义、全员参与的会议评审、项目和性能评估期间的评审与讨论。
制度化已定义的过程 (GG3) 共性实践： GP3.1 GP3.2	为在整个企业内阐明和了解敏捷价值观提供指南。通过使用回顾和收集过程相关经验，持续地改进和评估价值。	回顾包括收集与价值观相关的数据，例如过程相关的经验、技术经验以及提高团队性能的想法。 价值观得到明确定义并关联到相关框架和技术。	回顾记录、方针和价值观声明、培训材料。
集成项目管理： 为项目制定已定义的过程	敏捷团队可以利用 IPM 中的指南明确制定他们将为各自的团队或项目执行的过程，包括冲刺长度、发布持续时间、冲刺策划、待办事项梳理、回顾等。	敏捷团队从与敏捷价值观一致的过程资产集合中选择适当的框架和技术。	发布计划；冲刺计划、敏捷团队协议。

敏捷价值观（续）

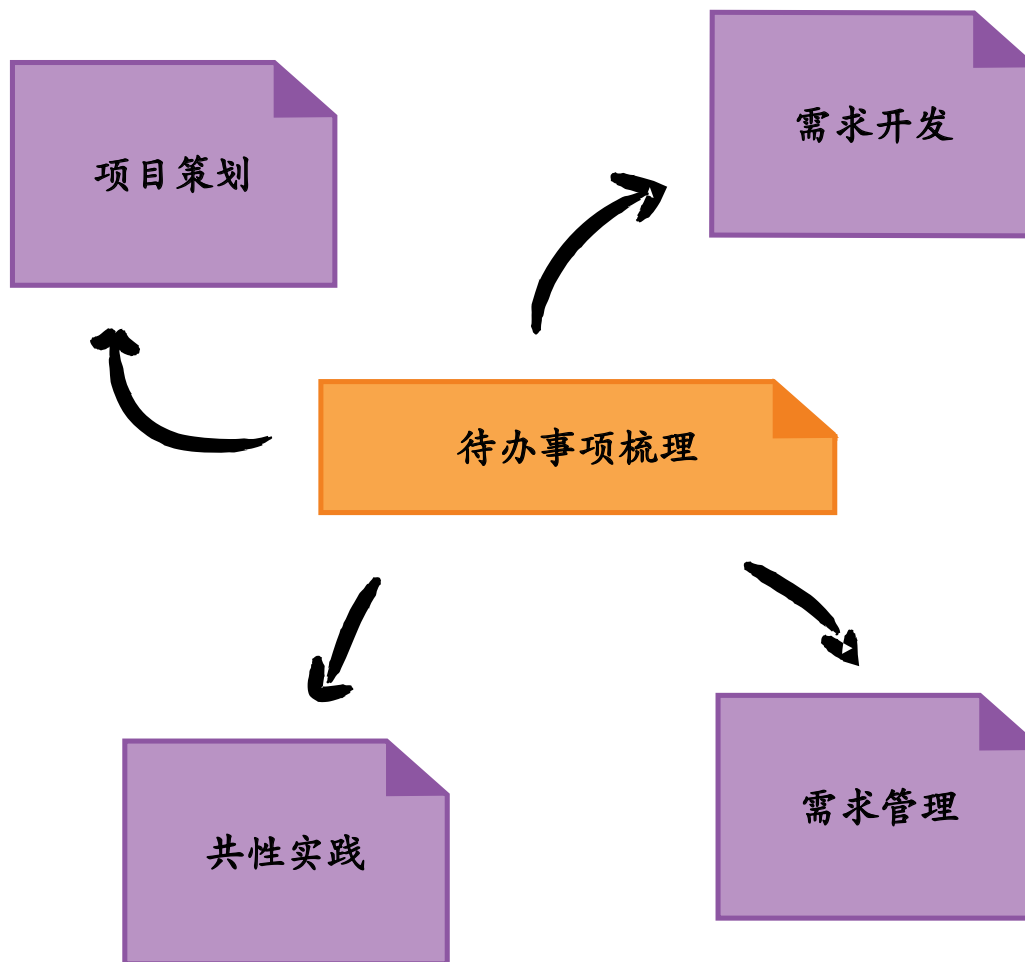
实践 / 目标	强化敏捷实施的方式：	能够符合实践 / 目标，前提是：	体现方式：
组织过程焦点 建立组织过程需求	CMMI 从敏捷过程需求的微观层面对整个模型提供指南。此指南可应用至所有实践和技术，以确保与敏捷价值观一致。	过程需求与组织内所采用的敏捷框架和技术一致。	然后，价值观路线图定义所采用的敏捷价值观和框架与技术之间的多对多关系。
组织过程定义 建立团队规则和准则	敏捷团队是敏捷价值观的主要消费者，CMMI 提供有关建立团队的准则，以便他们准备好全面采用组织定义的价值 观、框架和技术。	团队章程和团队协议用于定义与敏捷价值观一致的团队规范、过程、框架和技术。	团队协议、团队章程、团队合同。

待办事项梳理

“使用既定的准则创建或更改用户故事能强化冲刺待办事项列表和优化冲刺评审。”

待办事项梳理（有时称为“故事时间”）是 Scrum 团队使用的一种常见敏捷技术，以在冲刺之前和期间产生排好优先顺序的史诗和用户故事待办事项列表。在冲刺期间频繁发生的梳理有时被称为“微型梳理”。

待办事项梳理可从 CMMI 模型的各种实践中受益，其中需求管理、需求开发 and 项目策划是重点。为使待办事项梳理符合过程域中的具体实践，应该建立支持定期与客户或客户代表（产品所有者）交流的组织结构，经常与代表、业务 SME、客户 / 最终用户以及 Scrum 团队合作。



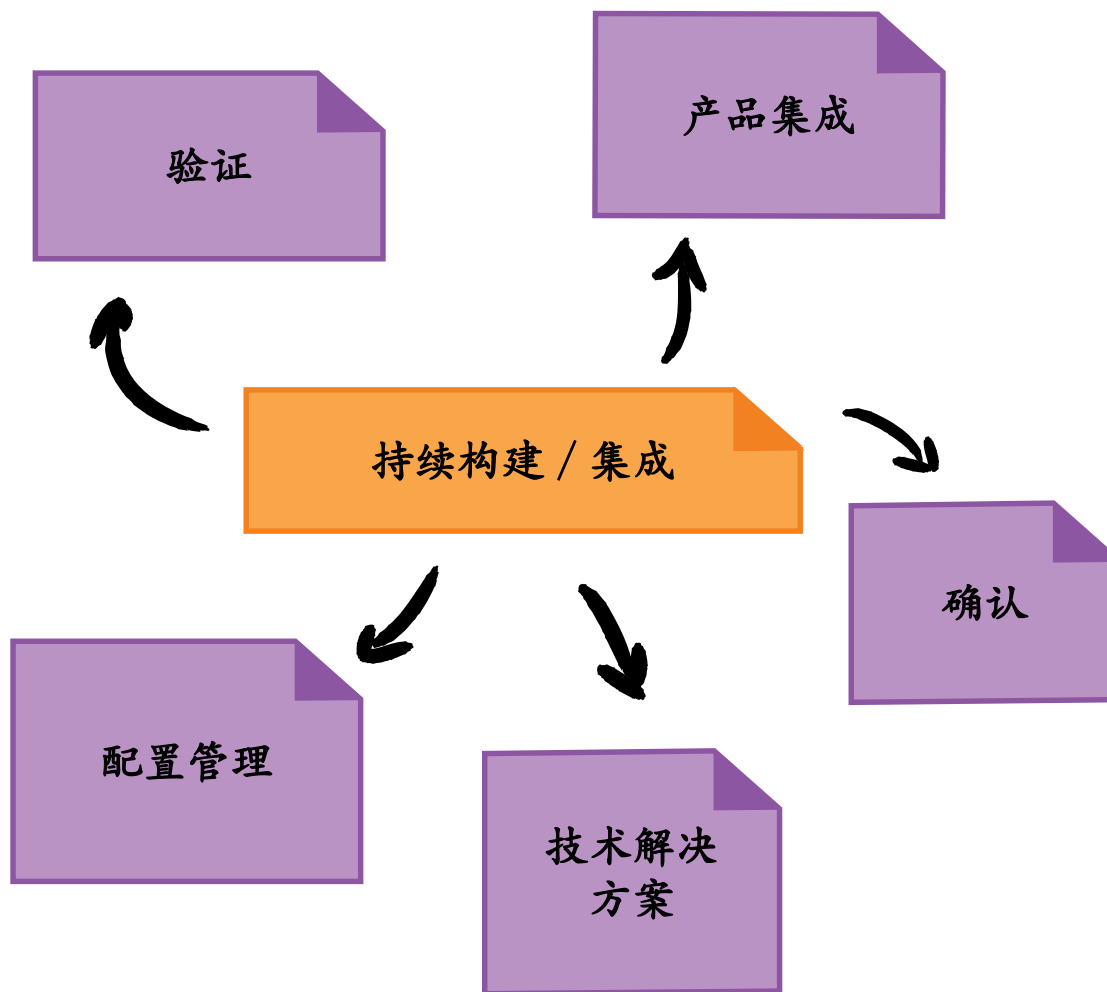
有关详细信息，请参阅第 52 页。

持续构建 / 持续集成

“项目在增量式阶段中组合产品组件时，使用目标明确的策略和已定义好的步骤，可加大成功集成的可能性。”

持续构建 / 持续集成是深受敏捷团队欢迎的持续测试和产品集成方法，在极限编程 (XP) 中首次引入。

CB/CI 可从验证、技术解决方案、产品集成和配置管理中包含的 CMMI 实践中获益。

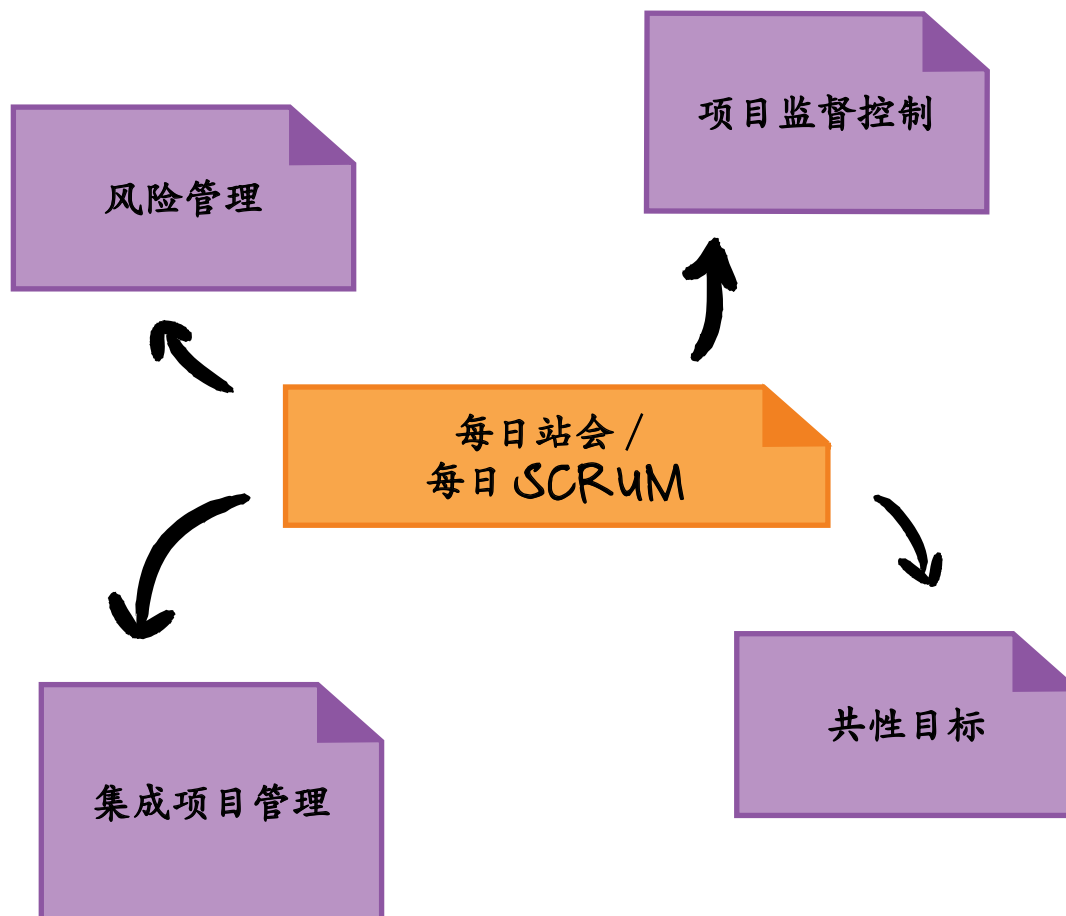


有关详细信息，请参阅第 58 页。

每日站会 / 每日 Scrum

“每日站会期间，团队经常会忽略的一个领域是风险识别与分析。”

每日站会（亦称“每日 Scrum”或“每日会议”）是常见于 Scrum 的一种敏捷方法，但也被广泛应用于所有类型的敏捷方法。与传统项目（“快速失败”）相比，它可更早发现问题和风险，以及加强敏捷团队成员之间的合作。每日站会展示了包括项目监督控制、风险管理、度量分析、集成项目管理中的实践，以及不同过程域的共性实践 GP2.8 和 GP2.10，具体视参与者而定。



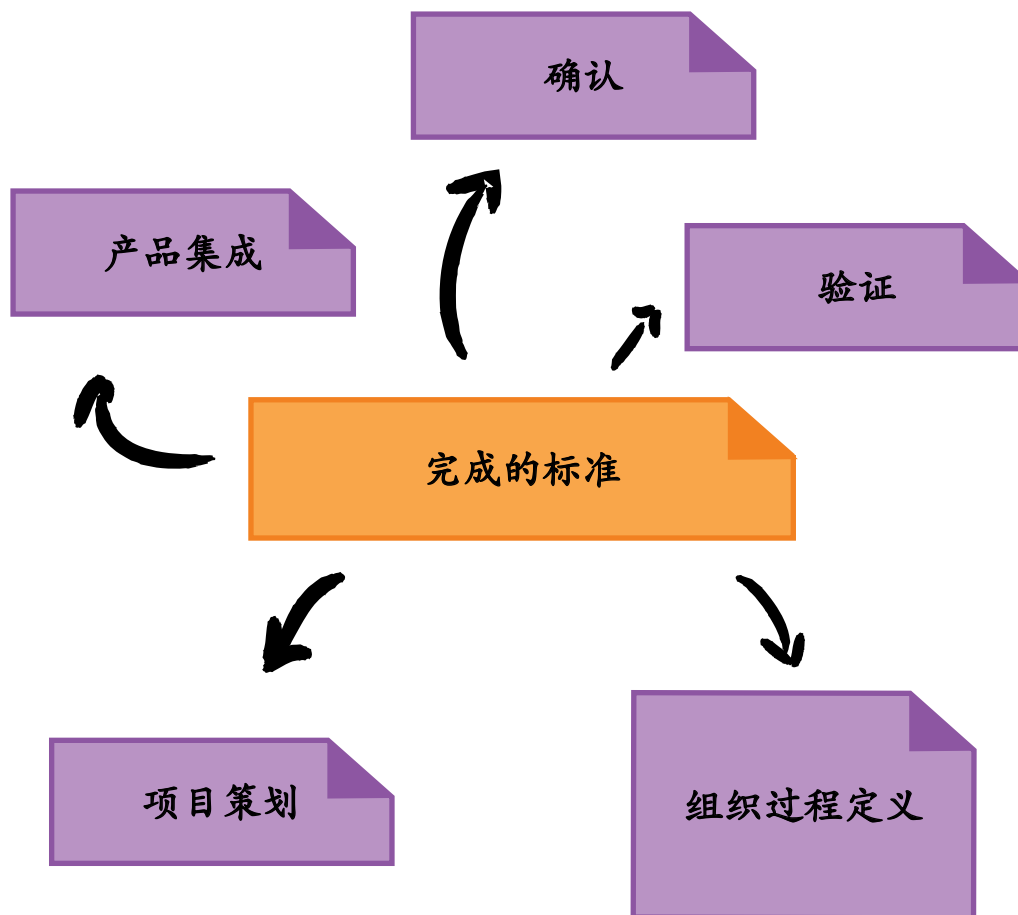
有关详细信息，请参阅第 62 页。

完成的标准

“完成的标准可能在单独应用到使用独特指标的史诗、用户故事和任务时更为有效。”

完成的标准 (DOD) 是任何敏捷项目的关键因素。它是敏捷团队内的一项协议，表明为产品准备妥当可供产品所有者进行冲刺评审必须完成的事项。对于大多数敏捷项目，“完成”的定义是在用户故事层面，由团队商定并在每个故事内定义。当完成故事内定义的 DOD 时，故事被视为“完成”。

DOD 是一个强大的工具，如果实施妥当，可以在尝试验证需求管理（特别是 SP1.1）、项目监督控制、确认、产品集成以及技术解决方案中的实践时，提供重要信息。



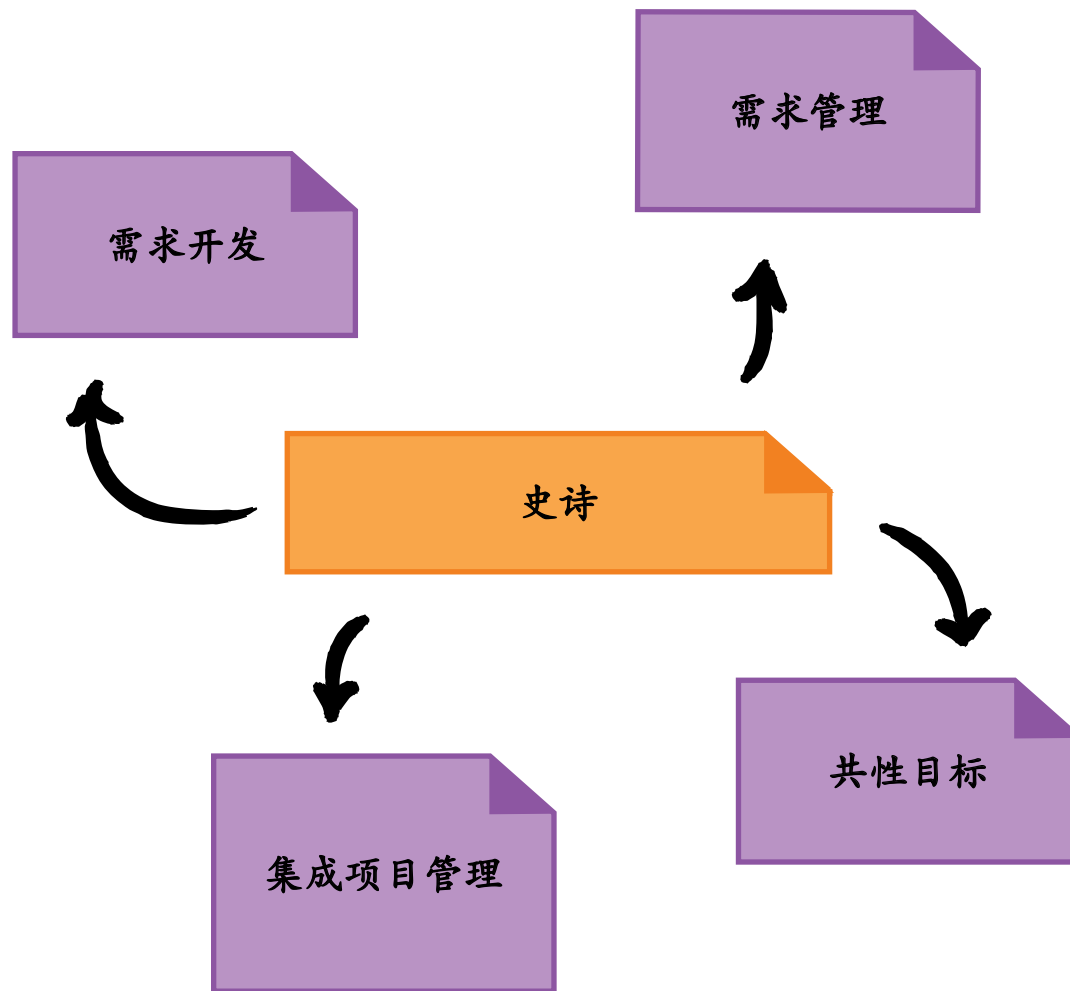
有关详细信息，请参阅第 67 页。

史诗

“因为史诗必须对应到其下属的用户故事，所以这构成第一级需求可追溯性。”

在敏捷术语中，“史诗”就是一个大型的用户故事，后面需要分解成独立的用户故事。通常，团队会选择将史诗分解成可在一个冲刺中完成的较小的、更易管理的用户故事。

史诗是作为需求获取和开发过程的结果创建的工作产品，因此与需求开发 (RD) SP1.1 和 RD SP1.2 密切相关。史诗可在联合开发研讨会 (JAD)、会议室模拟 (CRP)、头脑风暴研讨会中通过逆向工程或其他需求获取技术创建。



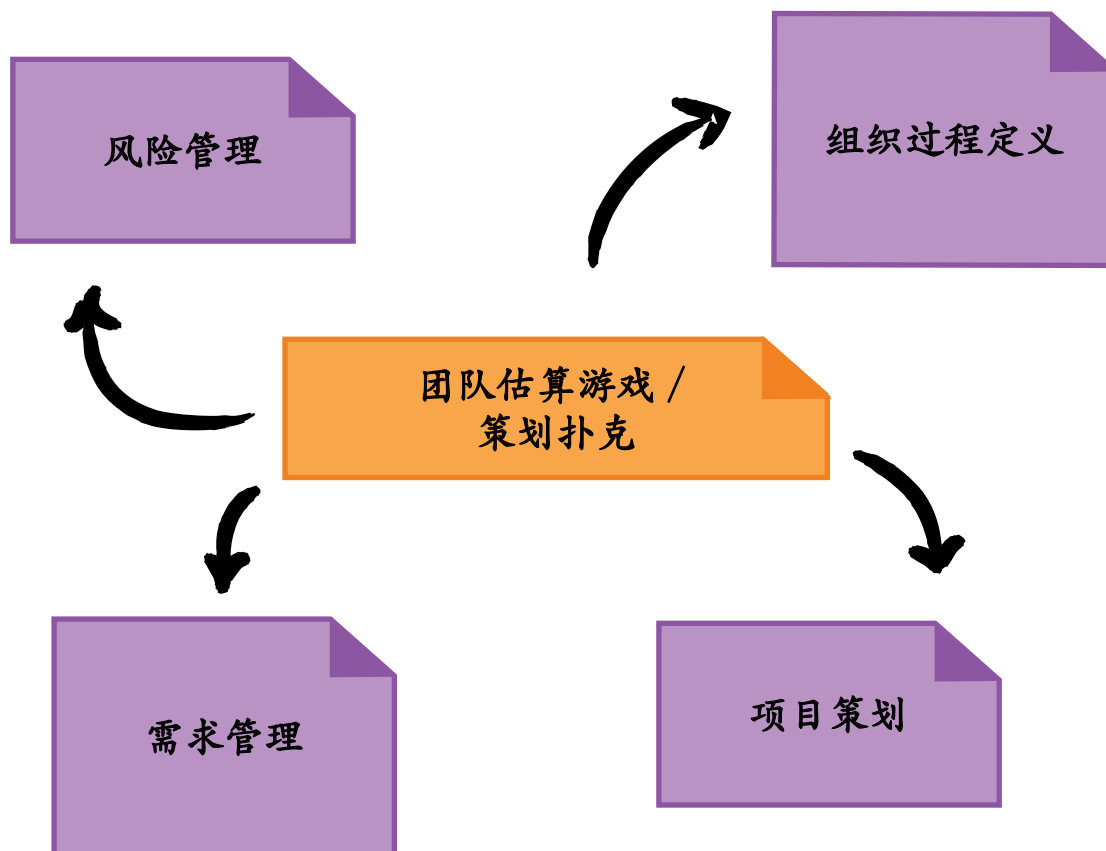
有关详细信息，请参阅第 69 页。

团队估算游戏（斐波那契游戏）与策划扑克

“团队估算游戏中，具备各种技能的团队成员将深入讨论每个用户故事的功能。”

团队估算游戏（有时也称为“斐波那契游戏”）是一项敏捷估算技术，它使用故事点和数量级粗略估算方法，估算相对规模。

在产品待办事项列表中，团队估算游戏主要反映项目策划 SP1.1 的使用，但前提是用户故事充实，包含任务和已经确定的“完成的标准”。

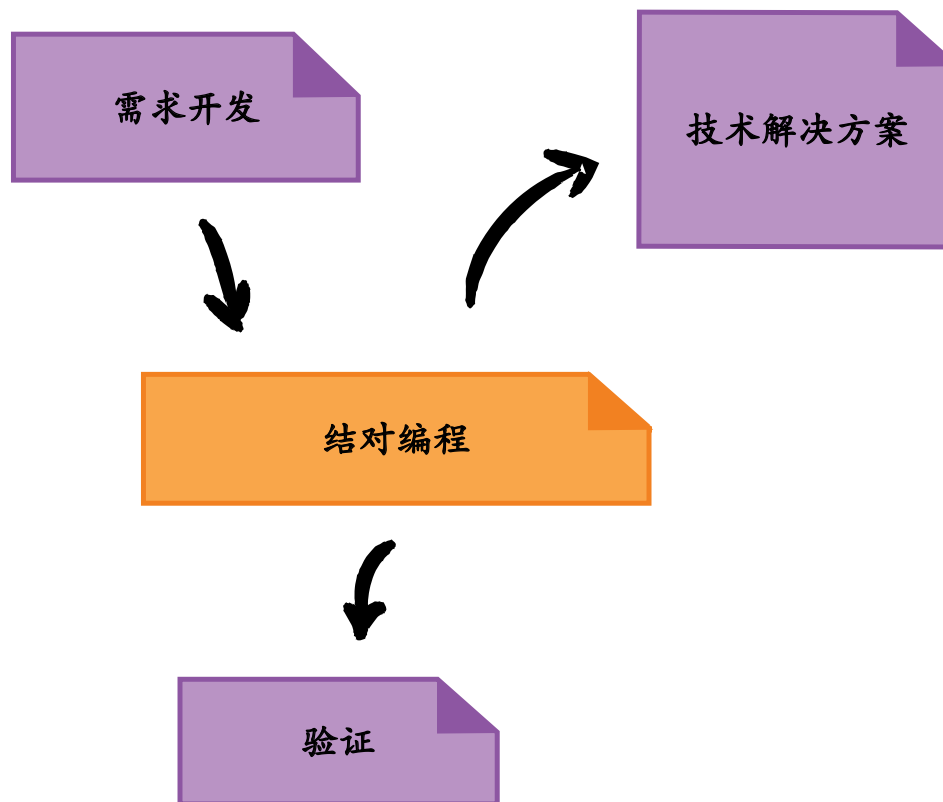


有关详细信息，请参阅第 73 页。

结对编程

“不管如何完成结对，但直接、即时的工作反馈环路让验证简单而有效。”

结对编程是一项敏捷技术，其中，两名软件开发人员合作完成编码任务。他们通常在同一个工作站工作：一个程序员为“驾驶员”，另一个为“导航员”，后者在前者输入代码时评审代码。虽然这通常导致编程的初始成本增加，但提高代码质量的回报更高。



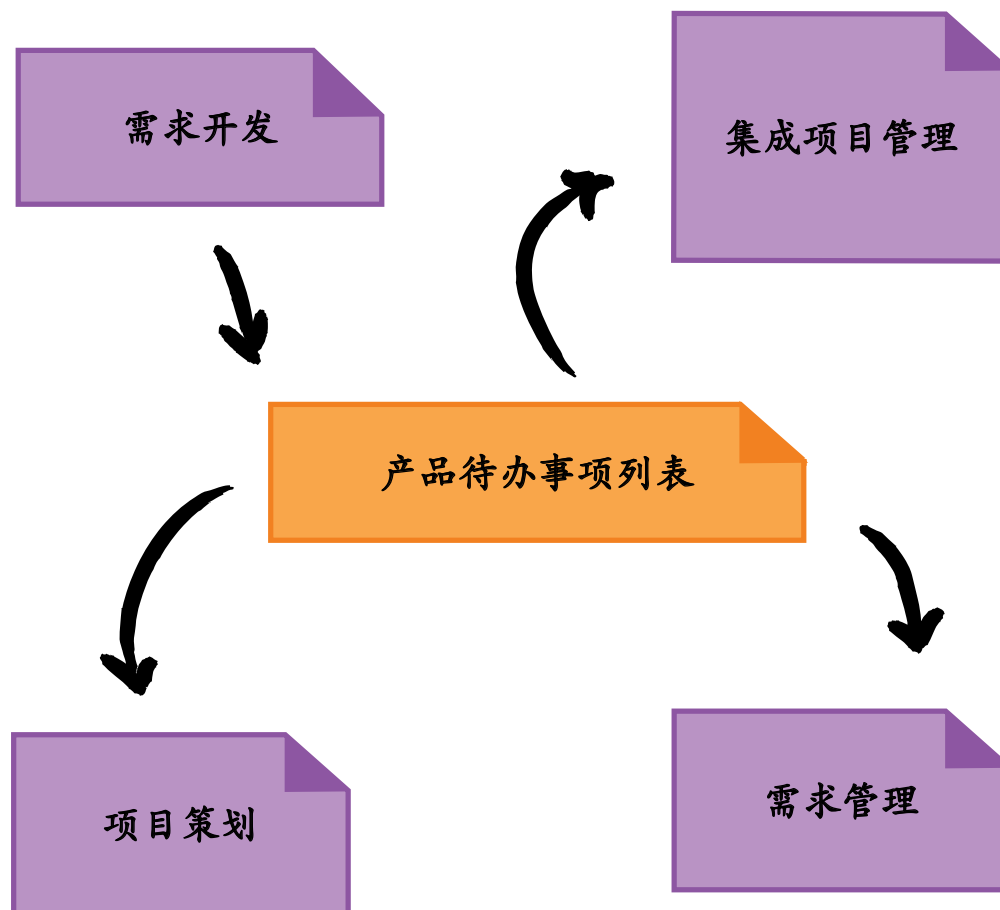
有关详细信息，请参阅第 76 页。

产品待办事项列表

“不管如何使用敏捷实践，产品待办事项列表是敏捷项目的基础。”

产品待办事项列表是产品所涵盖的一切事项的优先级列表。它包括特性、缺陷、文档变更和产品所有者要求的任何其他任务。

产品待办事项列表、用户故事、初始估算和优先级内容的建立，证明产品所有者了解并致力于满足产品需求。此数据也为项目计划的制定和相关计划的集成奠定基础。

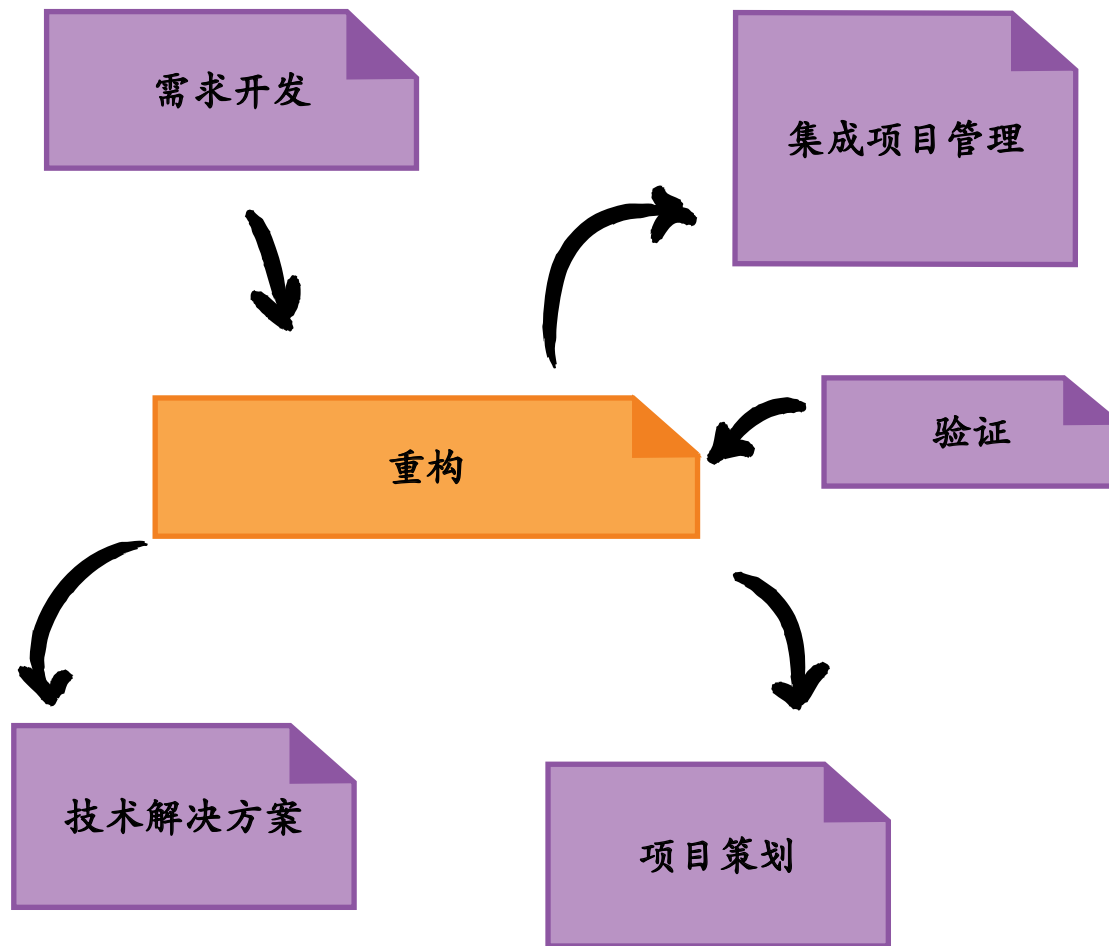


有关详细信息，请参阅第 81 页。

重构

“重构同样侧重于代码质量、需求质量、测试、集成或开发生命周期内的任何其他工作产品。”

重构技术侧重于改善软件代码的内部结构、风格、运算和条件，同时保持其外部功能和行为。重构侧重于持续改进产品，可以通过运用集成项目管理（SP1.7）、共性目标 3（GP3.2）和技术解决方案（SP3.1）中提供的准则改善。



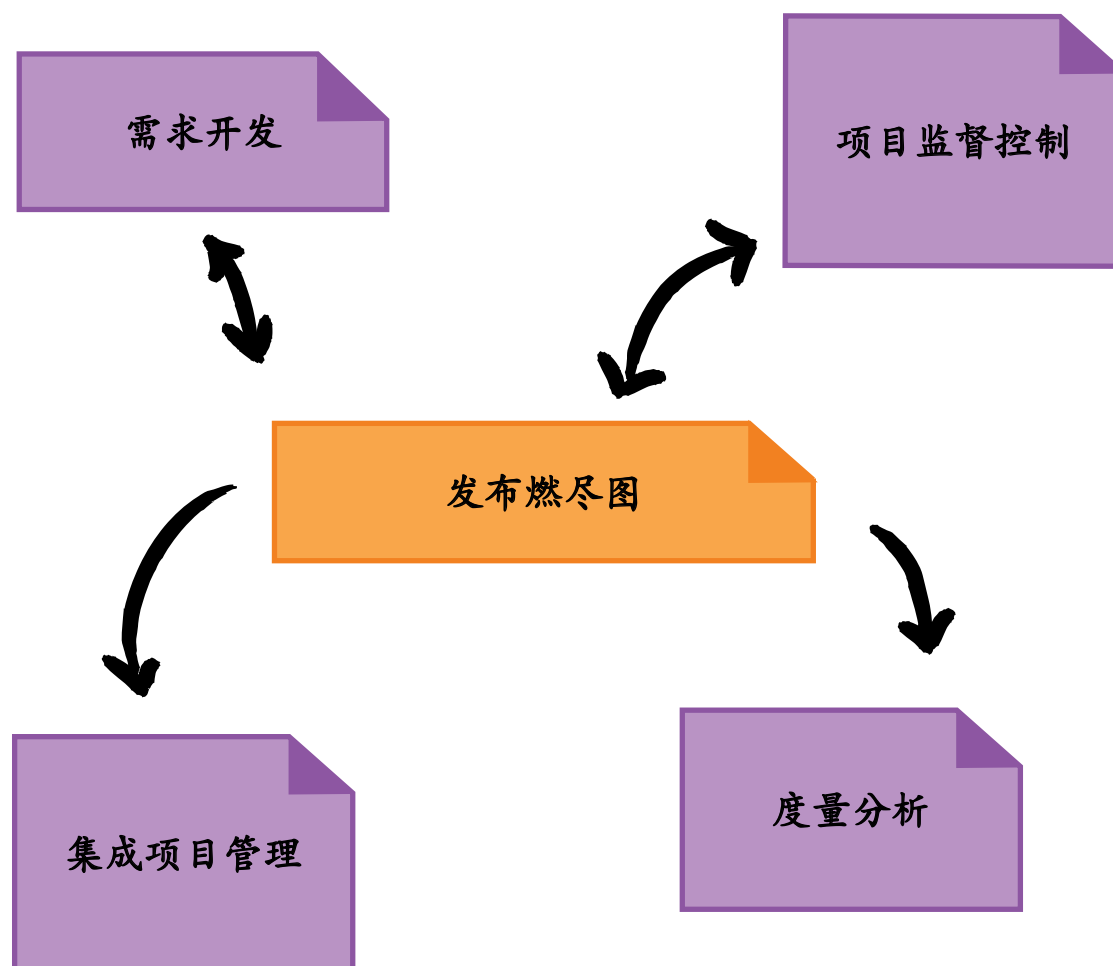
有关详细信息，请参阅第 85 页。

发布燃尽图

“发布燃尽图是为团队成员提供高级产品开发进度信息的信息辐射器。”

“发布燃尽图是按冲刺显示总体发布轨迹的信息辐射器。”

发布燃烧图是站在较高级别监控冲刺或迭代的发布实施进度的机制。冲刺燃烧图和每日站会跟踪冲刺期间的详细活动。

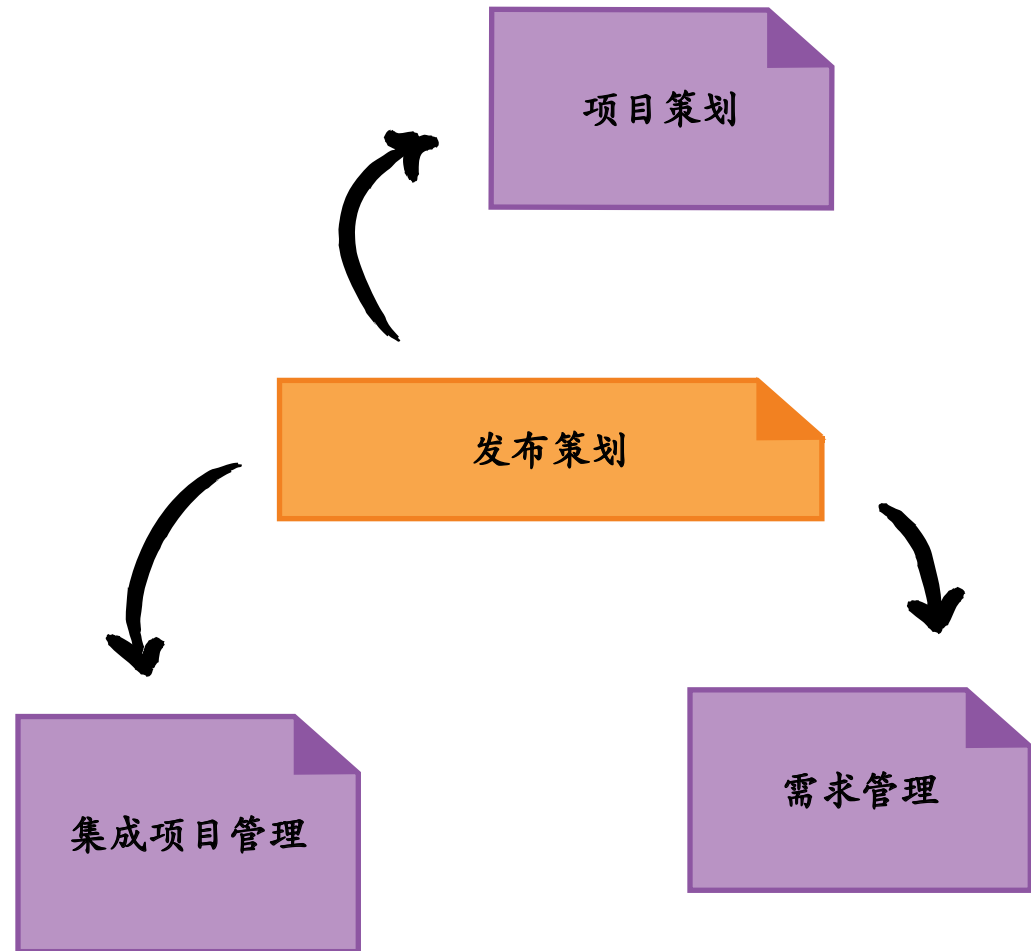


有关详细信息，请参阅第 87 页。

发布策划

“发布策划构成整个产品的基础，可能从 CMMI 在项目策划中提供的准则中受益。”

发布策划的目的在于制定交付产品价值增量的计划。发布策划期间，团队不会详细计划每项活动，而是推迟详细承诺至迭代 / 冲刺策划阶段，例如确定谁打算何时做什么事。

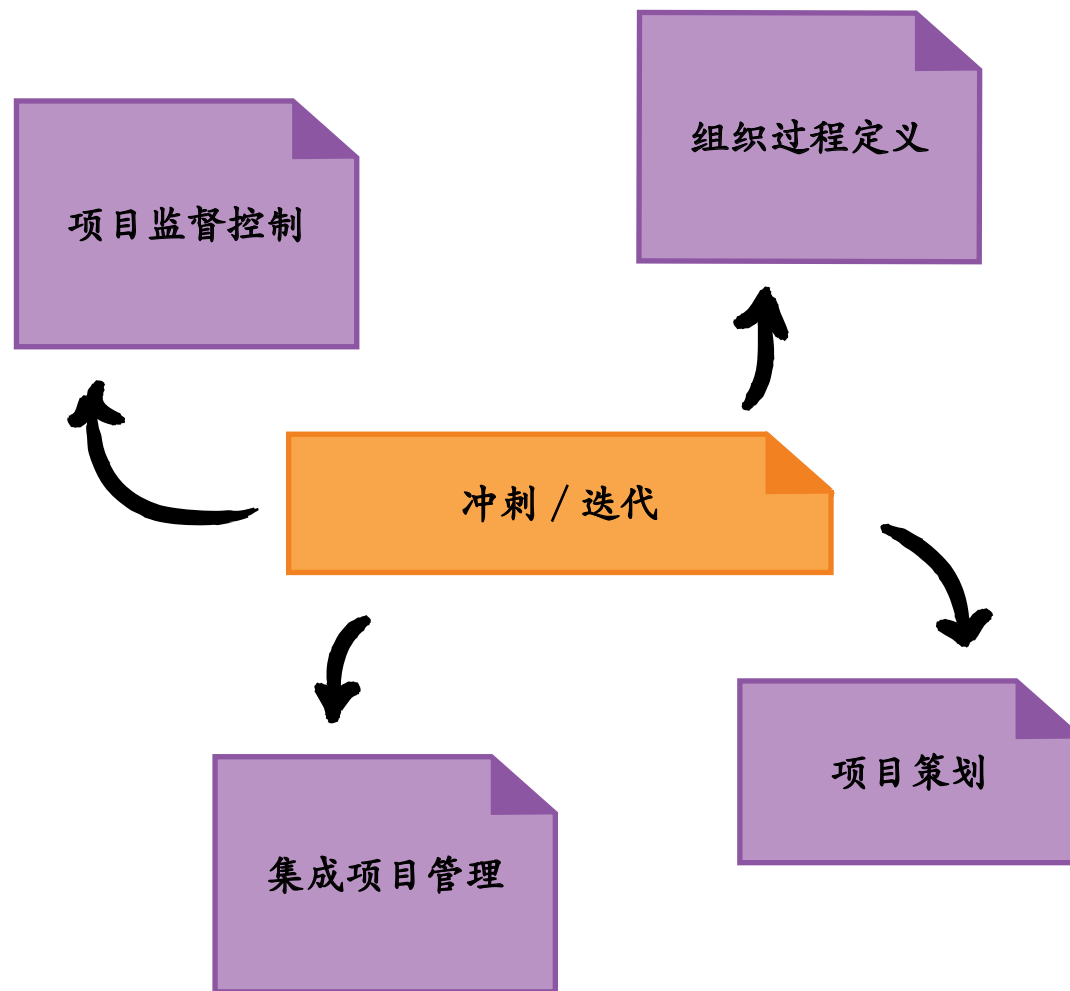


有关详细信息，请参阅第 91 页。

冲刺 / 迭代

“Scrum 团队可以试验冲刺持续时间并根据风险和团队经验等因素进行调整”

在敏捷团队中，完成任务的主要机制就是固定周期的冲刺。冲刺是一个时间箱的事件，团队成员在其中自行认领用户故事以及包括设计、编码、测试和待办事项梳理在内的所有工作。Scrum 团队在冲刺范围内包含的冲刺策划、冲刺演示和冲刺回顾各不相同，但都会在规定的时间箱内完成这些任务。



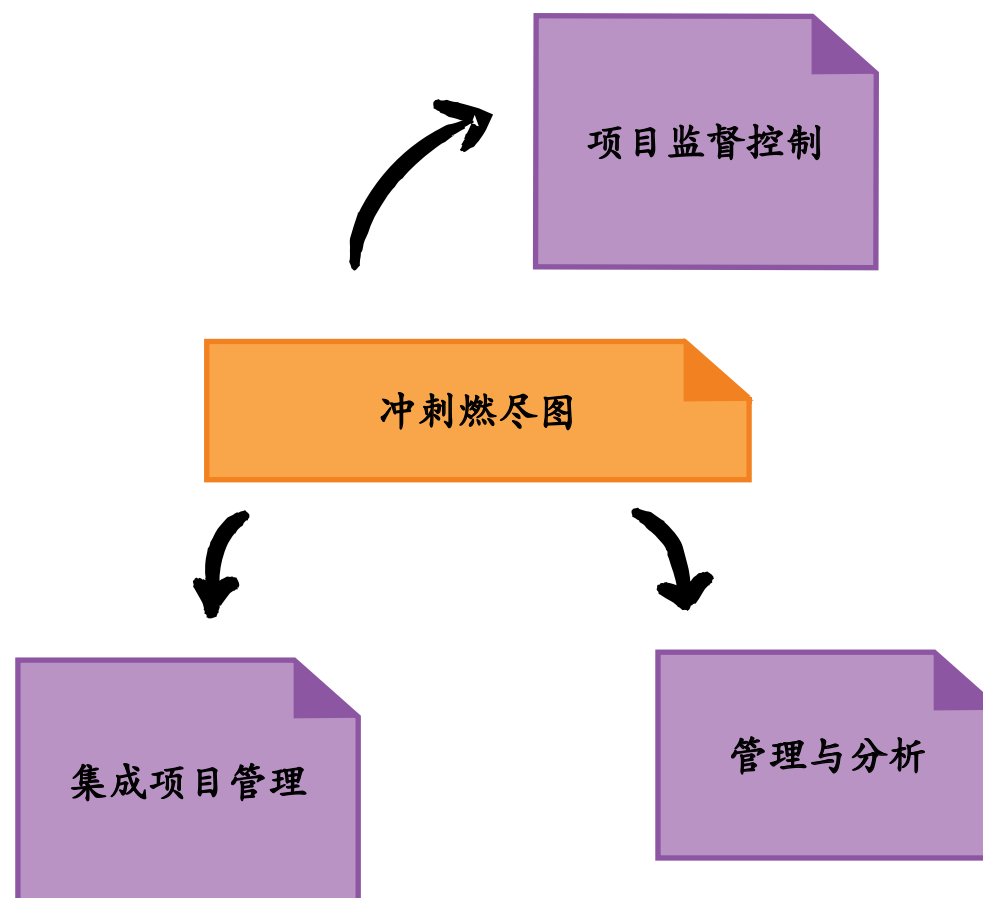
有关详细信息，请参阅第 96 页。

冲刺燃尽图

“评估团队可以从信息辐射器了解到很多信息，并应该在每个团队成员更新燃尽图时观察它们。”

冲刺燃尽图是直观描绘冲刺剩余工作量的信息辐射器。冲刺燃尽图帮助敏捷团队了解能否按计划在冲刺结束时交付预期功能。

冲刺跟踪中使用燃烧图的目的是监控团队工作进度以及管理完成迭代目标的风险，以交付潜在的可用产品。团队可以从 CMMI 的“度量分析”过程域中获益，提升自身的燃烧图分析能力。



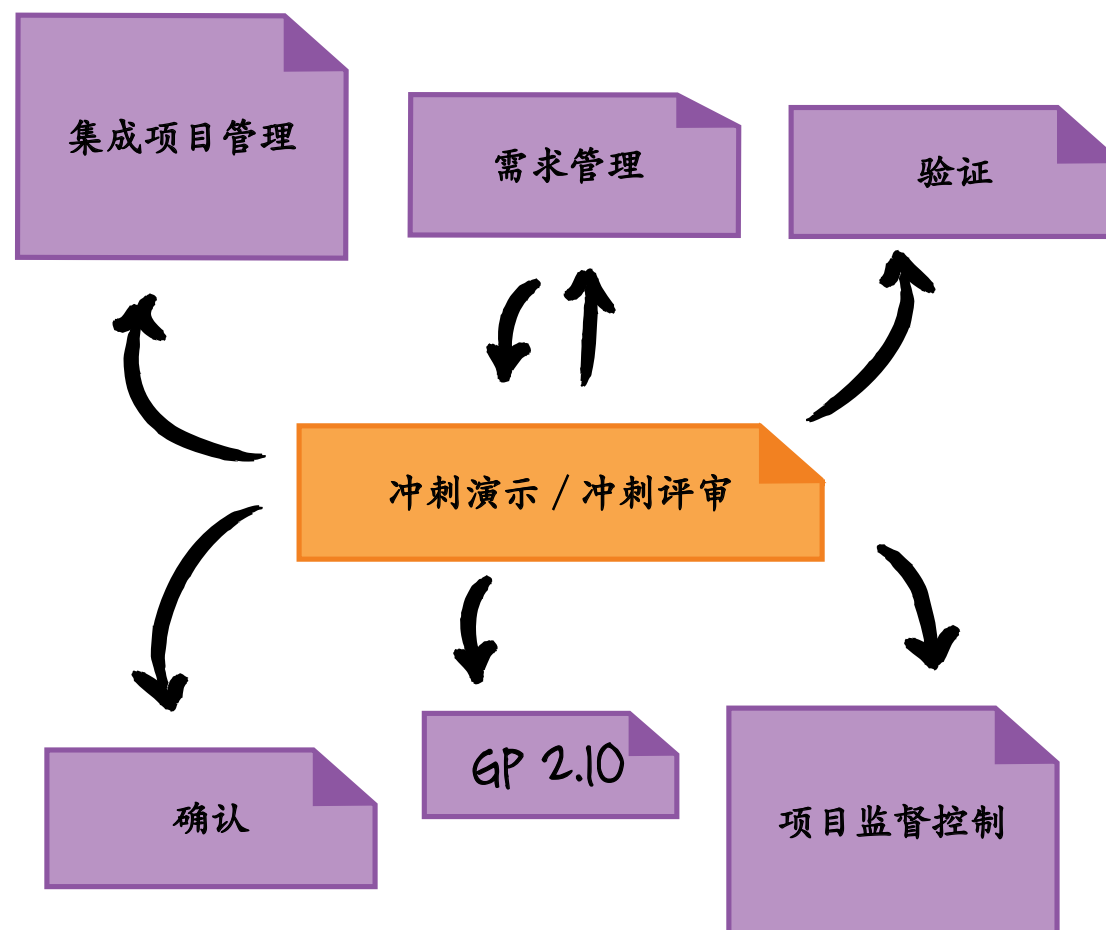
有关详细信息，请参阅第 98 页。

冲刺演示 / 冲刺评审

“关键利益相关方的参与对于冲刺演示的成功至关重要。”

冲刺演示（“冲刺评审”）是一种增量式迭代合作技术，可确保所有利益相关方清楚每个冲刺结束时敏捷团队交付的价值。

冲刺评审 / 冲刺演示可以通过运用项目监督控制、集成项目管理、验证和确认实践改善。



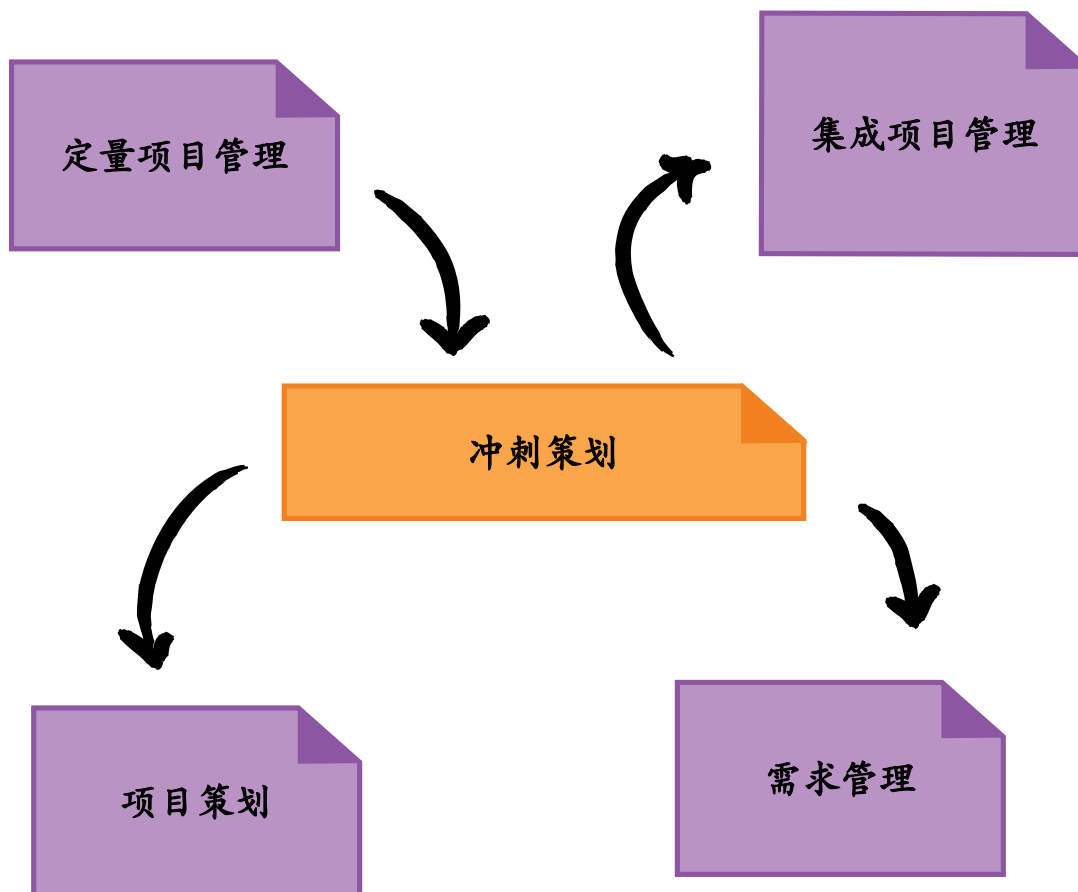
有关详细信息，请参阅第 101 页。

冲刺策划

“冲刺策划可以并且应该随着条件的变化和团队对于产品了解的加深，重新确定团队工作的优先级。”

冲刺策划会议在每个冲刺开始时召开，由敏捷团队和产品所有者（或客户）商讨将在即将开始的冲刺中交付什么价值。

在敏捷项目中，冲刺策划以迭代和增量方式频繁进行，并且会确认产品待办事项列表。



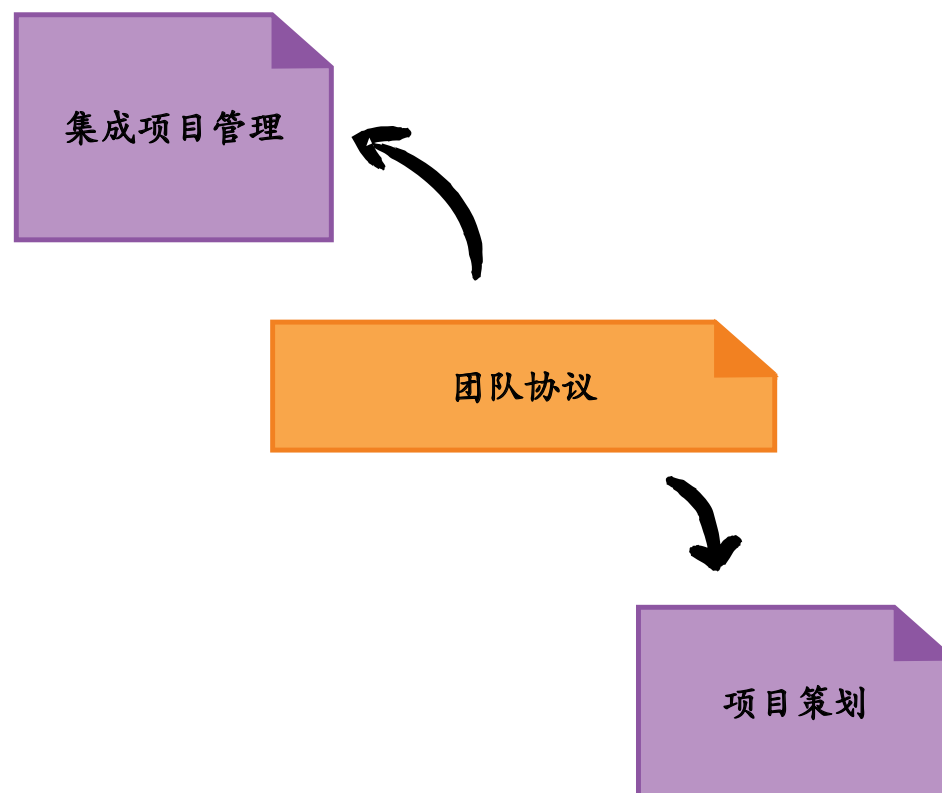
有关详细信息，请参阅第 107 页。

团队协议

“团队协议设定团队性能期望，代表每个敏捷团队的微观方针。”

团队协议是敏捷团队成员签订的社会契约，用于定义团队的行为、期望和达成一致的标准，有时也称为“工作协议”。

团队协议可通过运用集成项目管理和组织过程定义中的 CMMI 实践改进，其中包含有关组建团队的指南。



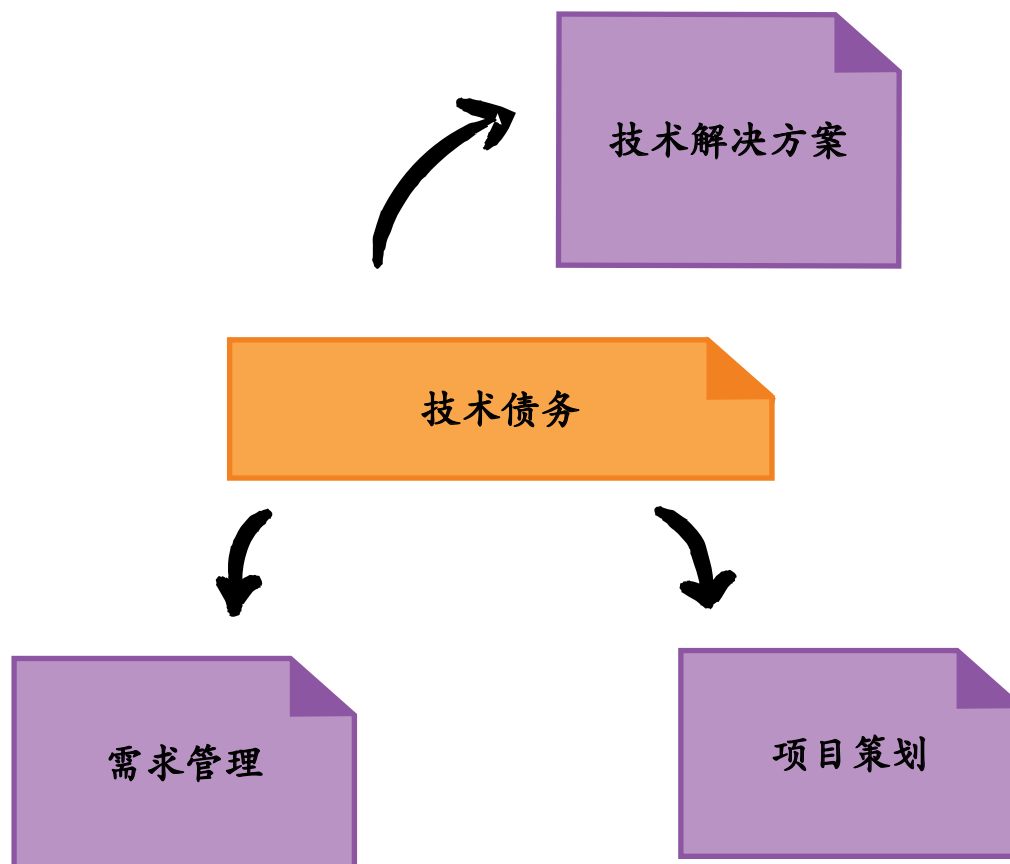
有关详细信息，请参阅第 113 页。

技术债务

“可以开发技术债务故事并将其置于产品待办事项列表，以分配到将来的冲刺中。”

当敏捷团队在时间、预算或资源的限制下主动决定采用不太理想、不太高效或不太可行的解决方案时，就引发了技术债务。

随着技术债务的增加，继续开发系统或维护现有系统的成本和工作量会变得过高，可在发布中预先安排“技术债务冲刺”以提高代码质量。



有关详细信息，请参阅第 116 页。

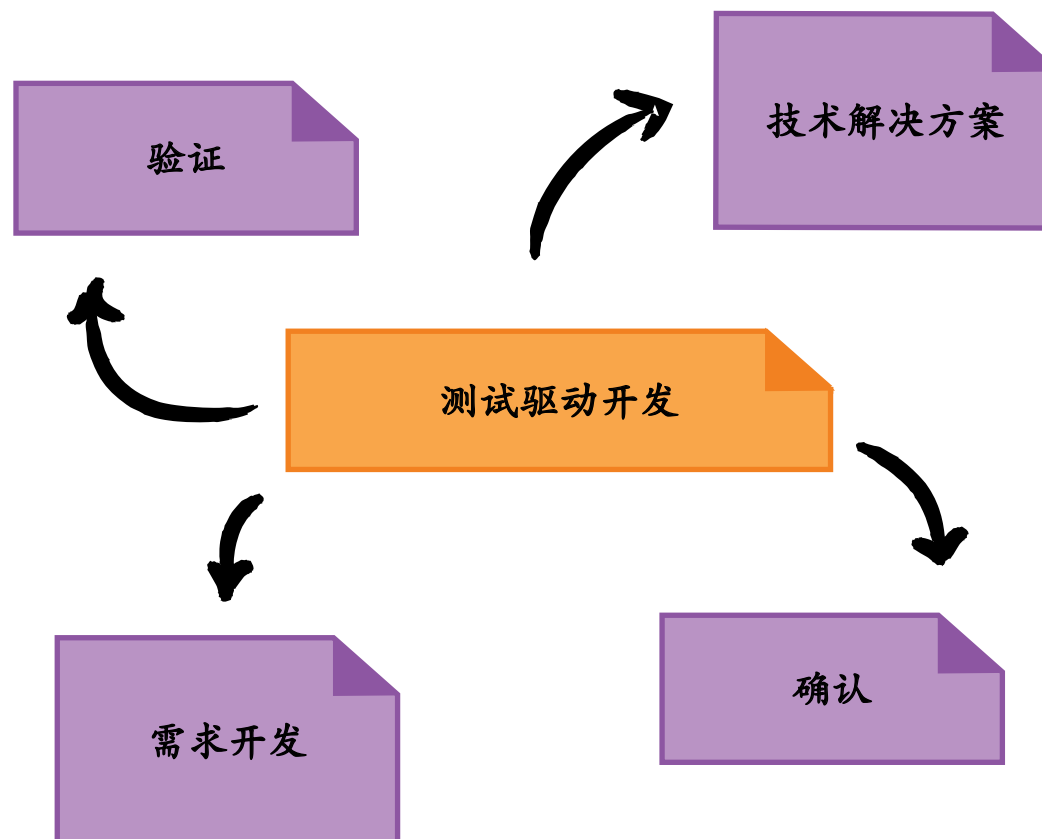
测试驱动开发

“TDD 可能是在编码之前将缺陷排除到产品之外的最佳机会。”

测试驱动开发 (TDD) 是极限编程 (XP) 项目中最常用的一种敏捷技术。

作为早期验证和确认的方法，一名开发人员将先编写一个基本的测试用例去验证所开发的功能；然后在知晓该测试将失败的情况下，编写最少量的代码来通过测试。接下来，开发人员将整理代码以确保其符合可接受的性能以及编码标准和原则。

测试驱动开发是可以改善代码质量和需求的强大方法，因此与验证、确认和需求开发有密切的关系。



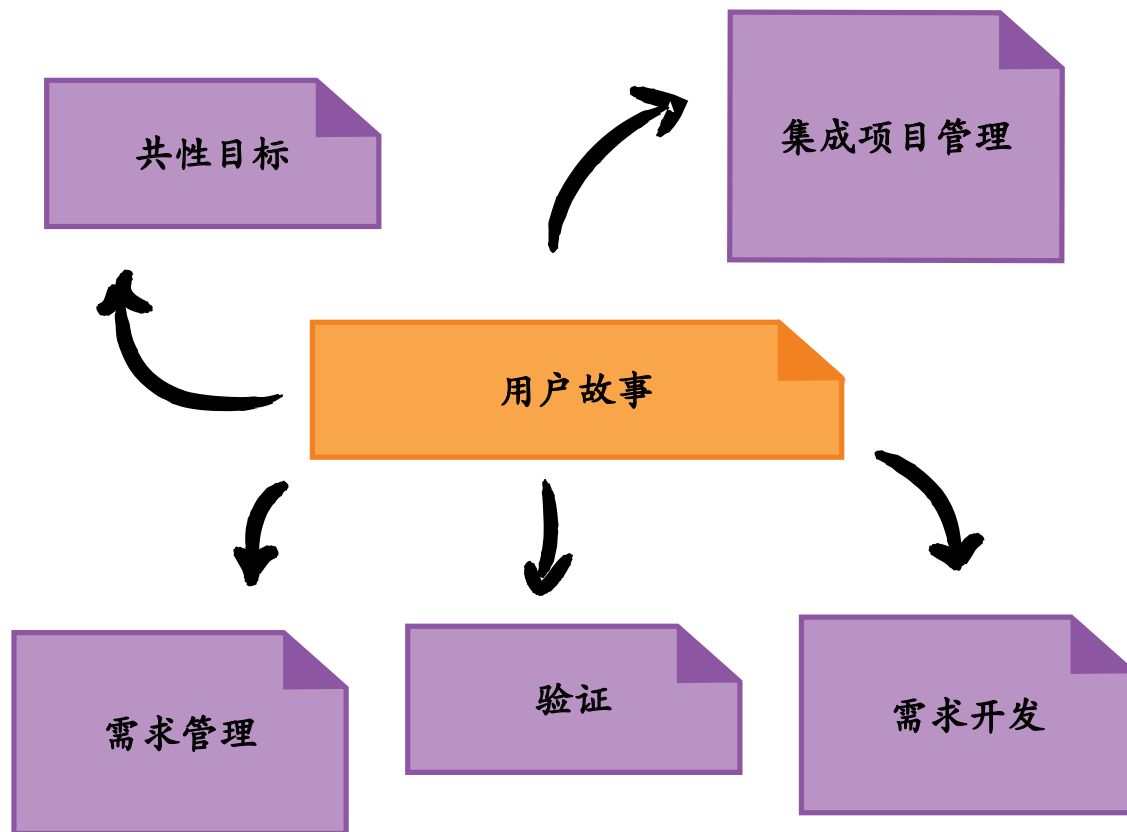
有关详细信息，请参阅第 118 页。

用户故事

“用户故事在用作包括待办事项列表、史诗、用户故事、子故事和任务在内的需求架构的一部分时，最为有效。”

用户故事是从用户的角度对离散的功能项所做的简单的语言描述。用户故事通常是从一个较大的“史诗”或用户故事细分下来的，它是可在单个冲刺中交付的独立功能项。

在采用测试驱动开发（TDD）并且严格定义了完成的标准（DOD）以界定故事的验收标准时，敏捷团队在需求开发过程中引入的缺陷会比较少。



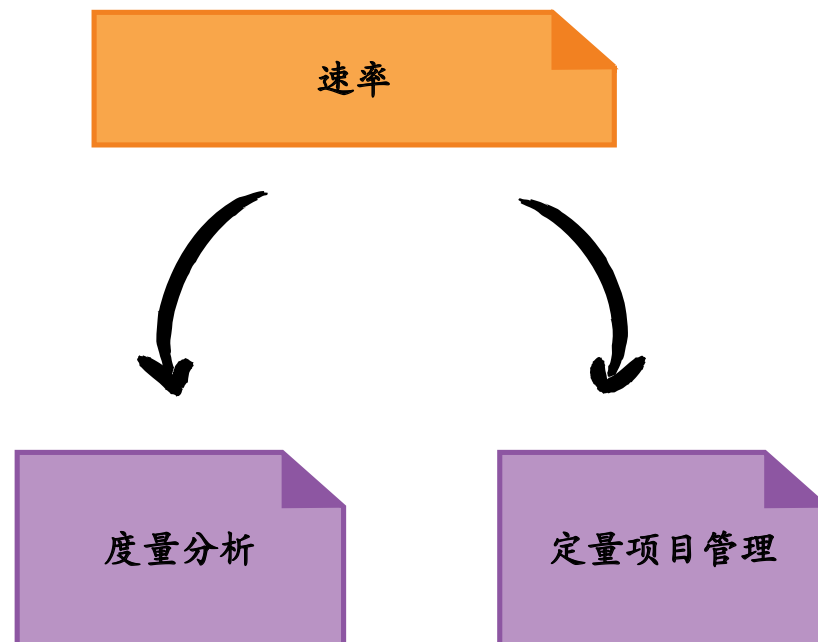
有关详细信息，请参阅第 123 页。

速率

“速率是敏捷的核心度量元之一，用于预测团队能否兑现其冲刺承诺。”

“速率”是敏捷团队在冲刺和发布期间用于能力策划和跟踪当前工作的工具。速率以达成共识的工作单位（通常是故事点）为基础，它表示了对于特定的 Scrum 团队在即将开始的冲刺中，预期交付多少价值或实际交付多少价值。

速率是敏捷团队的核心度量元，也是估算和跟踪项目工作的基础。因为敏捷团队是在“固定时间范围”、“固定团队规模”的环境中工作的，所以像 CPI 和 SPI 这样的度量元是多余的，速率才是衡量项目是否成功的重要度量元。



有关详细信息，请参阅第 126 页。

使用 CMMI 来增强敏捷成果

敏捷实践和技术全部可以通过适当运用 CMMI 加强。后续几页内容将介绍最常用于敏捷软件开发环境的实践。

Scrum of Scrum

Scrum 看板 / 任务板

测试驱动开发

策划扑克

持续构建 / 集成

冲刺 / 迭代

冲刺策划

冲刺燃尽图

冲刺演示 / 冲刺评审

重构

待办事项梳理

发布策划

发布燃尽图

价值

结对编程

开放空间技术

每日站会 / 每日 Scrum

史诗

速率

团队估算游戏

团队协议

完成的标准

用户故事

待办事项梳理

概述

待办事项梳理（有时称为“故事时间”）是 Scrum 团队使用的一种常见敏捷技术，用来在冲刺之前和期间产生排好优先顺序的史诗和用户故事待办事项。在冲刺期间发生的梳理有时被称为“微型梳理”。

待办事项梳理通常包含产品所有者与 Scrum 团队之间就将要为每个冲刺添加、删除或修订的史诗或用户故事进行的协商。所有利益相关方都会对此合作决策提供意见。因此，它是涉及冲刺策划和执行的一项关键活动。

产品所有者是产品待办事项列表的首要所有者。新的史诗和用户故事可以作为来自其他业务 SME、Scrum 团队、客户 / 最终用户的输入结果出现。产品所有者负责在产品待办事项列表以及相关验收标准中记录这些史诗和用户故事。

使用 CMMI 强化待办事项梳理

CMMI 可以引导团队成员朝着更好、更强的方式努力，生成经过优化的、具有完善用户故事的待办事项，从而加强待办事项梳理。如果待办事项梳理的目的是为冲刺策划澄清信息并做好准备，则需求管理 (REQM) 提供一个框架，来确保敏捷团队可以使用为每个故事或史诗确定的“完成的标准”商定、理解和管理其需求。

使用创建或更改用户故事的既定标准将强化冲刺待办事项列表和优化冲刺评审。每个用户故事使用陈述明确、完整、与其他故事一致、唯一识别、适当、可测试、可追踪、可实现且优先顺序确定的统一结构，将减少故事开发期间产生的缺陷。强大的变更管理工具和过程将使待办事项梳理和冲刺评审更强大，提供需求的可追溯性以提高测试完整性并简化维护，同时使故事中的变更与即将开始的冲刺的工作内容一致。

开发由“需求开发”过程域指导并确认的需求架构可以澄清和加强客户需求、史诗、用户故事、任务和测试用例之间的关系。此架构对于敏捷估算技术的使用至关重要，例如使用聚类规模法、策划扑克或团队估算游戏。

CMMI 主任评估师的观点：待办事项梳理

待办事项梳理可从 CMMI 模型的各种实践中受益，其中需求管理、需求开发和项目策划是重点。为使待办事项梳理符合过程域中的具体实践，应该建立支持定期与客户或客户代表（产品所有者）交流的组织结构，经常与代表、业务 SME、客户 / 最终用户以及 Scrum 团队合作。

评估团队应该能够验证待办事项梳理在整个冲刺期间迭代发生。团队应该会看到现有史诗和用户故事的优先顺序持续在变，因为用户故事优先级降低或产品路线图发生变化而从待办事项中删除用户故事的情况并不鲜见。

如果存在与更改产品路线图相关的统一说明，且相应的利益相关方使用“完成的标准”阐述用于接受和最终化用户故事的标准，则在待办事项梳理期间会展示需求管理。这通常在发布策划会议中完成，在该会议中将评审并根据客户和业务的反馈修改产品愿景。在此实践期间，Scrum 团队也将发现新的所需的故事。

需求管理 (REQM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
了解需求	对于撰写明确的、可实现的用户故事而言，提供需求的评价和验收准则是至关重要的。对于成功划分这些故事的优先级和实现这些故事，强调确保可理解也是至关重要的。	待办事项梳理如期进行；产品所有者、Scrum 教练和团队成员出席；故事和史诗在待办事项中；提出澄清型问题让相关人员理解用户故事或史诗的规模大小和复杂程度；用户故事“完成的标准”表明符合验收标准	出席待办事项梳理会议；查看待办事项详细目录；故事点已分配并且史诗调整完成；待办事项已记录在看板上或工具中
获取对需求的承诺	CMMI 侧重于收集反馈并确保在整个项目生命周期中与关键及广义利益相关方进行对话。识别广义利益相关方并监控他们对交付需求的参与情况，将有助于避免误解需求的含义。	待办事项梳理如期进行；所有成员都出席；新增的、修改的或删除的用户故事和史诗都经过每位成员的评审	出席待办事项梳理会议；查看待办事项详细目录；故事点已分配并且史诗调整完成；待办事项已记录在看板上或工具中
管理需求变更	CMMI 侧重于捕捉变更目的和历史。这一过程在敏捷组织内可能非常不正式，以致于接近混乱状态。有些变更结构确保交付最重要的商业价值。	使用 Jira 或 Version One 等工具将史诗和用户故事与任务以及完成的标准关联起来，或是采用手动关联	出席待办事项梳理会议；存在从史诗至父用户故事、子用户故事的映射（可能在某个工具中）；待办事项记录在看板上或工具中
维护双向可追溯性	双向可追溯性帮助确保完全实现了已确认的史诗和故事并识别仍然存在的差距。用于支持和确认与发布燃尽图相关的度量数据。	使用 Jira 或 Version One 等工具将史诗和用户故事与任务以及完成的标准关联起来，或是采用手动关联	出席待办事项梳理会议；存在从史诗至父用户故事、子用户故事的映射（可能在某个工具中）；待办事项记录在看板上或工具中
确保项目工作和需求之间的一致性	CMMI 提醒我们不断评审承诺、在用的实践以及它们如何支持用户故事实现。此实践也会满足回顾。	待办事项梳理如期进行；所有成员全部出席；用户故事得到澄清，然后与需求相关的期望确定所需的团队技能、所需环境、所需架构等	待办事项梳理会议召开；出席待办事项梳理会议；待办事项记录在看板或工具中

需求开发 (RD)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
获取需求	此实践强调积极从业务、客户、SME、团队和其他地方获取需求。年度产品规划、发布策划以及待办事项梳理共同作用，帮助符合此实践要求。	待办事项梳理会议是持续的，并且可以作为预定会议召开，以识别被记录为史诗和用户故事并包含在待办事项中的新增的和变化的需求	所记录史诗和用户故事记录在墙上或工具中；史诗（业务需求）和用户故事之间的映射记录在每个用户故事中；用户故事记录在看板上和工具中
将利益相关方需求转换为客户需求	此实践完全支持史诗和用户故事的创建。此实践包含了待办事项的优先级，它定义了实现的约束条件。	父与子用户故事在定期安排的待办事项梳理会议上创建、修改或删除	用户故事记录在看板或工具中；父与子用户故事显示在墙上或工具中
建立产品和产品组件需求	此实践提供架构、接口、非功能性需求及派生需求的背景。这些会被捕捉为用户故事。	因产品所有者的变更对用户故事和史诗进行的修改对应了本 SP 的“维护”方面的内容，“需求的建立”是通过创建包含实现该需求所需的更多具体数据的父或子用户故事来满足的	用户故事记录在看板或工具中
确定接口需求	此实践要求将接口需求记录为用户故事	接口需求通过用户故事识别和解决	接口用户故事记录在看板或工具中
建立操作概念和场景	此实践支持定义有关故事如何实现的用户场景（业务场景）以及这样做的必要性。这在用户故事环境下应该是显而易见的。	待办事项梳理会议会创建用户故事和史诗，阐述用户故事的书写方式必须使用格式“作为 XXX，我需要 XXX 以便能 XXX”。	用户故事包含于待办事项中，显示在看板或工具中；为了说明如何用于实现业务需求，用户故事使用了标准格式明确阐述业务场景。

需求开发 (RD)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立所需功能和质量属性的定义	此实践支持定义故事“好到什么程度”。功能的逻辑分组支持理解故事的顺序，这对于确保交付一个完整的功能系统至关重要。	待办事项梳理期间填入的用户故事包含定义“成功是什么样子”的验收标准和质量属性，并用于定义测试和用户故事展示验收标准	验收标准融入用户故事。测试脚本实现了验收标准。演示和冲刺评审确认达到验收标准。
分析需求	此实践侧重于确保所有史诗和用户故事都是必需的，并且支持更高级别的需求（父 / 子关系）。这也满足需求跟踪。	待办事项梳理会议期间，产品所有者和团队分析史诗和用户故事并提出澄清型问题，以了解实现它们的必要条件	在待办事项中更新用户故事和史诗以反映分析结果；出席待办事项梳理会议
分析需求以实现平衡	此实践侧重于根据优先级、规模和工作量确保故事的平衡。这可能有利于促进创造“预研”或实验性故事，侧重于原型设计、模拟或概念验证。	待办事项梳理会议期间，用户故事优先顺序列为 1 到 n 以确保首先实现最关键的故事。这是冲刺策划的输入信息，冲刺策划将进一步平衡所执行的工作。	史诗和故事记录在待办事项中并且优先顺序列为 1 到 n
确认需求	此实践确保史诗和用户故事满足最终用户需求。这可以通过预研用户故事来完成；如果基本故事格式提供有关故事将如何使用的推理（如果编写无误则应该提供），也可以从基本故事格式完成。这也与待办事项梳理的其中一个关键目的相符。	待办事项梳理会议期间，将讨论确认需求（故事）的方法并达成共识，并讨论为原型、模拟和展示所创建的故事。	出席待办事项梳理会议；确定记录了待确认的故事（原型、模拟和展示）

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算项目范围	此实践帮助团队重点了解范围所需的数据集。虽然实践要求“WBS”，但它通常以非传统形式表达。在排定好优先级的待办事项中的史诗 / 故事，同时也包含定义每个故事“就绪 / 完成的标准”以及规模的元数据，它们构成传统 WBS 的等效功能，故事被分配给某个冲刺或迭代后尤其如此。	待办事项梳理会议期间，确定史诗和用户故事规模。对于史诗，使用 T 恤码号表示；对于用户故事，使用故事规模表示（通常使用策划扑克或等效方法）	出席待办事项梳理会议；评审史诗和用户故事并确保规模数据显而易见
确定工作产品和任务属性的估算	此实践要求团队对工作进行估算。这在发布和冲刺策划中完成，但也可在确定待办事项史诗和故事规模时完成。在待办事项梳理期间，这通过创建父与子故事确认。实践中的其他因素可以帮助团队考虑创建类似复用因素之类的代码。	待办事项梳理会议期间，使用故事规模确定父与子用户故事的规模（通常使用策划扑克或等效方法） 故事任务通常通过冲刺策划管理，但很多组织会在用户故事中包含标准任务	出席待办事项梳理会议；评审父与子用户故事并确保规模数据显而易见；根据组织采用的过程评审任务级数据的故事
估算工作量和成本	此实践确保组织同时估算工作量和成本。在敏捷开发中，工作通常受限于时间和人员，所以成本需要考虑人数和时间。实践鼓励考虑新技术以及需要增加多少故事来获得所需的结果。	在待办事项梳理会议期间，故事的规模用于估算完成具体任务所需的时间（参见冲刺策划），待办事项梳理期间应该使用在冲刺策划期间使用的相同技术	出席待办事项梳理会议；评审史诗并确保提供规模信息
控制工作产品	此实践确定 CM 是策划过程的关键组成这一事实，并且代码和故事版本的维护对于了解历史和未来需求十分重要	史诗和用户故事有序存储并映射到相应的史诗；父子关系显而易见，并且更新受控	可能使用了工具或其他适当的机制来维护史诗和用户故事版本
识别利益相关方并使其参与进来 (GP2.7)	此实践确定了确保识别所有利益相关方并让其参与生命周期适当部分的必要性。其好处在于这是一项经过计划的并应在待办事项梳理期间讨论的深思熟虑的活动。	产品所有者、主题专家和团队成员出席待办事项梳理会议以确保一致性。	出席待办事项梳理会议；确保所有相应的利益相关方都在场；可能会使用工具或其他适当的机制维护史诗和用户故事的版本

持续构建 / 集成

概述

持续构建 / 持续集成是深受敏捷团队欢迎的持续测试和产品集成方法，在极限编程 (XP) 中首次引入。CB/CI 可用于任何类型的软件环境并且不限于敏捷团队。在 CB/CI 环境中，每当有新代码“提交”至代码管理系统时，都会使用自动化工具构建一个应用程序，并进行单元测试或集成测试。它最初是与“测试驱动开发 (TDD)”技术一同引入的，后者同时撰写测试用例与用户故事。

使用 CMMI 强化持续构建与集成

用于验证、确认和配置管理的 CMMI 实践描绘通向成功的 CB/CI 体验的路线。为 VAL/VER 选择产品、建立 VAL/VER 环境以及制定程序和标准对于成功搭建 CB/CI 环境都必不可少。这些实践确保了持续构建可以节省时间，并且可以重复执行任务，并逐渐增加可演示的功能。自动化的结果和警报将为分析提供重要信息以逐步改进产品并有助于整体开发工作。

CMMI 的产品集成实践支持持续构建环境的迭代式和增量式特性。当项目使用目标明确的策略和已定义好的步骤分阶段增量式组装产品组件时，成功集成的几率就大大提高。

使用 CMMI 实践引导 CM 基础架构的开发与改进，将加强和支持不断变化的、持续或每日构建的环境。一旦完成开发，CM 实践支持的清晰、一致的行为文化将提高产品基线的完整性。

CMMI 主任评估师的观点：持续构建 / 持续集成

CB/CI 实践需要代码库 (任何配置管理系统的主要组件) 以及服务器和脚本，可以自动 (以频繁的时间间隔) 收集新提交的代码、构建和编译系统，运行自动化的单元与集成测试，然后公布结果。CB/CI 的扩展将构建的系统自动部署到测试甚至生产环境 (此扩展不常见)。CB/CI 可从验证、技术解决方案、产品集成和配置管理中包含的 CMMI 实践中获益。

验证 (VER) / 确认 (VAL)

实践	强化敏捷实施的方式:	满足实践要求的条件:	体现方式:
建立验证 / 确认环境; 执行验证 / 确认	对于哪些工作产品应该实施 CB/CI, 提供指南和案例	开发人员必须经常 (例如每天) 提交他们的代码; 必须频繁 (例如每次提交时或定期) 运行自动化的构建和测试; 必须将结果提供给开发人员和测试人员; 必须明确在每个冲刺期间处理的用户故事和任务中的哪些模块将成为 CI 组件。	撰写了自动化单元与集成测试用例; 使用了 TDD 方法 (测试驱动开发); “完成的标准” 包含了在用户故事中的测试用例, 有时可在工具或看板的便签条中找到; 将冲刺待办事项中的用户故事追溯到任务和模块的某种方式
建立验证 / 确认环境	提供验证与确认环境的准则	自动化的构建服务器用于构建和运行单元与集成测试; 已安装并运行相应的服务器以支持 CB/CI; 支持 CI 所需的环境已定义并实施 — 模拟器、仿真器、场景生成器以及与其他系统交互的接口。	自动化构建和测试活动的结果已公布, 例如发布在网页或监视器上; CB/CI 实施的架构文档; 持续构建服务器; 测试用具
建立验证 / 确认程序和标准	促进对构建事件确认步骤的清晰、文档化的了解	执行 CB/CI 的步骤对团队成员来说清晰明了; 编写脚本以自动化执行过程。	已编写用于自动化执行过程的脚本; 对于那些中断构建的问题, 修复速度的准则
分析验证 / 确认结果	给利益相关方提供指南, 并为已发现的缺陷进行参数化定义。	CB/CI 已自动化执行并且无需人工干预 CB/CI 工具生成有关结果的报告和数据, 并且该数据已被自动发送至对应的团队成员	已编写脚本 已安装硬件来支持 CB/CI 已安装软件来支持 CB/CI 自动化构建和测试活动的结果已公布, 例如发布在网页或监视器上
分析验证和确认结果	参见上文	CB/CI 工具生成有关确认结果的报告和数据, 并且该数据已经自动发送至相应的团队成员以供评审。报告和数据的分析将为 PMC 计划安排提供输入, 可能的纠正措施以及可能需要的进一步活动。	构建报告 通过工具生成的电子邮件 缺陷报告

产品集成 (PI)

实践	强化敏捷实施的方式:	满足实践要求的条件:	体现方式:
建立集成战略; 建立产品集成环境	为开发战略和产品集成环境提供方针、利益相关方和计划的准则	CB/CI 系统 (自动化构建、单元与集成测试) 已部署到位并在使用中	实施 CB/CI 实施的架构文档; 构建脚本; 自动化构建和测试活动的结果已公布, 例如发布在网页或监视器上
确认用于集成的产品组件已就绪	提供有关集成确认的准则	CB/CI 系统 (自动化构建、单元与集成测试) 已部署到位并在使用中	构建脚本; 对于那些中断构建的问题, 修复速度的准则
评审接口说明的完备性	提供有关接口确认的准则	CB/CI 系统 (自动化构建、单元与集成测试) 已部署到位并且在使用中; 当新代码提交时, 假设脚本已完成可进行接口确认	自动化构建和测试活动的结果
评估组装后的组件	提供有关接口确认的准则	CB/CI 系统已到位并在使用中; 在整个开发生命周期执行整套脚本	自动化构建和测试活动的结果已公布, 例如发布在网页或监视器上



每日站会 / 每日 Scrum

概述

每日站会（亦称“每日 Scrum”或“每日会议”）是常见于 Scrum 的一种敏捷方法，但也被广泛应用于所有类型的敏捷方法。与传统项目（“快速失败”）相比，它可更早发现问题和风险，以及加强敏捷团队成员之间的合作。每日站会一般时长 15 分钟，收集来自每位团队成员的信息，侧重于以下四个关键领域：

- 我已经完成了什么（“昨天的天气”）
- 我将要做什么
- 我遇到什么阻碍

在团队圆桌会议期间，成员们共同探讨，以确定所发现的问题和风险的最佳解决方式。

Scrum 将出席者标识为“Scrum 团队”或核心团队成员，但建议不要让客户或管理层代表参与。在实践中，出席会议的团队成员往往会增加。通常，参与完成当前冲刺的每个人都会参加会议。

团队可能使用某种记号来指示目前允许发言的人。这确保一次只有一人发言，并且整个团队都会听到，不会受到干扰。

进行每日站会的方法有两种。默认方法是在可以基于用户故事听进展状态的地方召开站会。

另一种方法称作“轮询调度”，用于进行与不受用户故事限制的任务相关的讨论，例如缺陷管理、持续改进等。以用户故事为中心的讨论的益处在于焦点从个人切换到故事，与把每日站会转变成没有任何益处的状态报告会正相反。

使用 CMMI 强化每日站会

项目监督控制（PMC）实践为很多敏捷实践（包括每日站会）提供阐述和指导。CMMI 对加强监控活动、确保问题解决、消除进度计划偏离等方面有明确要求。每日站会提供了一个对等机会进行进度汇报、计划活动汇报并识别阻碍。PMC 中的实践可以帮助此实践侧重于正确的事情：监控进度、承诺及风险。当识别出问题时，可以在开放、高效的环境中沟通、分析并设法解决问题。

每日站会期间，团队经常会忽略的一个领域是风险识别与分析。使用 PP、PMC 和风险管理 (RSKM) 中的实践，敏捷团队可以实施一个简单的过程来确保团队正采用适当级别的风险管理措施。每日站会期间，Scrum 教练应该负责记录属于预定义类别或来源的风险，可能采用用于某些估算技术中的相同 T 恤码号进行度量。

如果团队认为每日站会不重要并且低效，就可能开始跳过该会议。如果对每日站会没有纪律或时限要求，其他团队可能会拖延。集成项目管理 (IPM) 提供良好的实践来解决此问题。建立团队工作环境将确保所有团队具备有效开展每日站会所需的设施和工具。这可能会定义

用于涵盖远程团队成员的工具、具有大量看板的 Scrum 室以及敏捷管理工具。IPM 还针对建立团队提供指导准则。谁会出席每日站会？何时开始、如何运行以及如何记录决策和活动？IPM 实践还为 Scrum 团队提供了一些在管理项目的利益相关方问题、依赖关系以及协调问题解决方面的指导。

CMMI 的共性实践对于任何敏捷环境的成功都至关重要。CMMI 的共性实践，虽然通常被视为“常识”，但也往往被忽视。提供资源以及有关如何执行项目的培训和指导对于获得可预测的一致结果也至关重要。

CMMI 主任评估师的观点：

每日站会 / 每日 SCRUM

每日站会展示了包括项目监督控制、风险管理、度量分析、集成项目管理中的实践，以及不同过程域的共性实践 GP2.8 和 GP2.10，具体视参与者而定。

因为每日站会具有协作、非正式的特性，所以评估团队应尽量在评估期间观察每天召开的这些会议，并且主任评估师应该清楚非正式工具，例如白板、Scrum 看板或看板以及便签条，这些工具可帮助团队成员监控和跟踪其工作。

项目监督控制 (PMC)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
监控项目策划参数	此实践鼓励应用每日站会来监控每日进度以及使用标准的敏捷度量元。这还可以促使团队在遇到阻碍时确定是否具备所需要的技能、良好定义的故事等。	每日站会如期进行，所有成员都会出席，由每位成员评审用户故事和任务的状态	出席每日站会；查看多个用于收集状态的 Scrum 看板或工具；显示剩余故事点和工作量的燃尽图；冲刺目标以及发布目标完成数据
监控承诺	此实践需要使用冲刺燃尽图和 ETC	每日站会如期进行，所有成员都出席，采用“我做了什么、我在做什么、我遇到什么阻碍”的结构化格式，或遵循类似的其他方法	出席每日站会；查看多个 Scrum 看板；发布燃尽图、冲刺燃尽图变更及剩余工作量；工具（例如 Jira）中的日期
监控项目风险	此实践将帮助确保发现团队风险，并且团队就此立即展开讨论或将风险记录下来供后续跟进。它还鼓励在看板或工具中记录风险。	具体讨论并记录风险。将风险作为特定议程项目在特定站会上讨论，而不只是作为当前的一个阻碍因素	风险在信息辐射器（例如 Scrum 看板）上标出；风险列在工具（例如 SharePoint、Jira）中；风险标注在用户故事上；风险讨论当日在 Scrum 看板上张贴记录
开展进度评审	此实践与每日站会的目的完全一致。它支持会议召开、度量数据收集、风险识别和阻碍收集。	每日站会如期进行，所有成员都出席，采用“我做了什么、我在做什么、我遇到什么阻碍”的结构化格式，或遵循类似的其他方法	每日站会是每日衡量团队进展的举措。信息辐射器（燃尽图、Scrum 看板等）提供其他信息。
开展里程碑评审	此实践支持使用结构化的事件，以在项目的关键点了解所取得的成绩。尽管此实践大多数情况下被当成冲刺评审，但敏捷团队有时会在冲刺的最后一次每日站会上举办“冲刺结束”庆祝活动。	已执行团队评审或回顾来标记冲刺和发布结束	在冲刺和发布总结时，记录评审或回顾中提出的建议和计划调整

集成项目管理 (IPM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
使用集成计划管理项目	此实践帮助团队侧重于使用整个团队统一使用的过程，并支持可以根据团队收集的数据改进的一套标准敏捷实践。此实践同样支持需求管理和针对发布计划的反思。	团队根据几乎每日与产品所有者确认的冲刺计划和优先级管理每日工作	出席每日站会； 评审冲刺计划并确认与站会中解决的故事的一致性
管理利益相关方参与度	此实践鼓励基于团队与利益相关方之间的合作将问题（阻碍）文档化。问题记录将是重要成果。	团队、产品所有者和其他受影响的利益相关方出席并参与；已识别依赖关系和任何风险 / 阻碍	出席每日站会； 观察与会者并确保团队在场； 依赖关系已标识在故事、Scrum 墙或工具中
管理依赖关系	此实践侧重于管理来自其他团队的关键依赖关系以及记录和监控这些依赖关系的需求。在敏捷开发中，会尽一切努力减少此工作，以使这类依赖关系要么不存在，要么被边缘化。一旦存在关键依赖，团队的深思熟虑以及认真监控至关重要。	团队将确保在故事内识别跨团队依赖关系并进行讨论	出席每日站会； 依赖关系已标识在故事、Scrum 墙或工具中

风险管理 (RSKM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
识别风险	此实践确保通过每日站会识别风险和阻碍。	专门讨论和记录风险及阻碍	信息辐射器，比如有便签条的看板； 工具，如 Jira 或 SharePoint
评估风险、对风险进行分类并确定优先级	此实践确保团队花费时间了解每项风险和阻碍，以便根据紧急性和重要性逐项解决，最终达成冲刺目标。	专门讨论风险和阻碍并确定优先级，并记录其类别和来源。	信息辐射器，比如有便签条的看板； 工具，如 Jira 或 SharePoint
制定风险缓解计划	此实践确保每项风险和阻碍不仅得到理解，而且已指定人员进行处理并制定解决计划（可能形成一个用户故事以专门解决该阻碍）。	专门讨论与记录了风险和阻碍及其缓解计划。	信息辐射器，比如有便签条的看板； 工具，如 Jira 或 SharePoint

完成的标准

概述

完成的标准（DOD）是任何敏捷项目的关键因素。它是敏捷团队内的一项协议，表明为产品准备妥当可供产品所有者进行冲刺评审必须完成的事项。对于大多数敏捷项目，“完成”的定义是在用户故事层面，由团队商定并在每个故事内定义。当完成故事内定义的 DOD 时，故事被视为“完成”。DOD 可能包括：

- 撰写代码和测试用例
- 代码经过同行评审（或采用结对编程生成）并符合开发标准
- 代码单元测试通过
- 用户故事测试通过
- 代码部署到系统测试环境并通过系统测试
- 代码部署到集成环境并通过集成测试
- 用户故事 / 测试用例通过 UAT（用户验收测试）。UAT 基于为用户故事 / 测试用例制定的验收标准。

- 任务剩余时间设为零，用户故事 / 测试用例已移至“完成”
- 已生成、评审并批准相关文档 / 图表，包括产品文档（用户指南、帮助等）、架构文档以及客户要求的任何其他文档
- 任何构建、部署、配置变更已经被落实、被文档化并被沟通了

敏捷项目的每个元素都服从于 DOD，包括多次冲刺策划、用户故事、冲刺演示、回顾和待办事项梳理。DOD 定义完成每个元素所需的任务和工作项。

CMMI 主任评估师的观点：完成的标准

DOD 是一个强大的工具，如果实施妥当，可以在尝试验证需求管理（特别是 SP1.1）、项目监督控制、确认、产品集成以及技术解决方案中的实践时，提供重要信息。各 DOD 的文档化说明可以为组织过程定义和集成项目管理中的实践提供验证。

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算项目范围	识别项目范围的所有相关组件	每个元素的“DOD”集合构成一个虚拟 WBS。因此，必须完成 DOD 以满足项目团队需求，在冲刺策划会议期间确切了解用户故事中所含内容，并且具备足够详细信息将故事点分配执行。	WIKI 中的标准 DOD；附加到用户故事或史诗；在冲刺计划中使用故事点值标识
确定工作产品和任务属性的估算	提供关于待估算工作产品包的指南	DOD 在冲刺策划期间定义，每项的工作量在冲刺的总体故事点值中考虑；必须结合策划扑克等敏捷估算技术	估算反映 DOD 中的任务；冲刺计划展示 DOD 的考虑事项

确认 (VAL) 和验证 (VER)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
执行验证 / 确认	提供验证与确认的准则	DOD 定义足够详细以验证每个相关工作产品的完成情况或确认每个用户故事	来自反映特定工作产品 DOD 的信息辐射器的任务完成指标；剩余工作量减至零

史诗

概述

在敏捷术语中，“史诗”就是一个大型的用户故事，后面需要分解成独立的用户故事。通常，团队会选择将史诗分解成可在一个冲刺中完成的较小的、更易管理的用户故事。

冲刺分配不是分解史诗的唯一原因。另一考虑因素是功能的明确性。客户要求的特性通常较为复杂，需要分解成多个小组件以便理解和实现。另一考虑因素是“谁将完成该工作？”如果用户故事要求多个 Scrum 团队来完成工作，则故事会被视为史诗，因为该故事将需要分成至少两个用户故事，每个 Scrum 团队负责一个。

史诗通常在待办事项梳理期间分解，有时也在发布或产品策划期间分解，或者由产品所有者持续分解。分解过程往往是迭代式的。产品所

有者可以请求一项特性并将其作为用户故事提供给团队。团队可以将此称为史诗，并进一步分解。敏捷参与者没有将用户故事称之为史诗的标准度量指标，虽然通常理解为史诗可以跨越多个冲刺，但用户故事不应该。

使用 CMMI 强化史诗 / 需求开发

史诗是大型用户故事，需要进一步细分后，才能分配给冲刺待办事项。有关需求的不确定性，特别是在项目早期，会推动概括地记录已知内容的需求，并促成敏捷项目中史诗的创建。但负责需求定义的利益相关方如何才能采取结构化的方法呢？敏捷并不意味着混乱，因此寻求 CMMI 指导准则是常识。

CMMI “需求开发”过程域中的特定目标和实践可以清晰地阐明创建史诗和用户故事的过程，还有助于提高产品待办事项列表的可靠性。以下是一些示例：

- 确保在创建产品待办事项列表时考虑所有可能的产品需求来源，而不仅仅是外部客户的需求。在项目的早期阶段遗漏重要需求会在后期产生巨大的负面影响。
- 确保相关的利益相关方参与史诗和用户故事的创建和评审过程，以减少冲刺中的误解和返工。
- 确保已分析和确认产品待办事项推动了冲刺策划。应该明确待办事项的测试或验证方法。

当反馈环路较短且团队成员不断进行经验教训总结时，敏捷团队才会成长。“需求开发”的说明性资料中提供了关于如何确保产品待办事项列表中的史诗和用户故事完整可靠的绝佳方法。

敏捷项目的目标之一是能够快速、有效地响应整个项目生命周期中不可避免会发生的变更。项目时间表变更、团队成员变更、信息需求变更，但最有破坏性的变更是需求变更。

Scrum 是一种轻型项目管理框架，其实施细节由采用该方法的团队来决定。CMMI 的“需求管理”过程域提供许多最佳实践，可帮助敏捷团队在进行产品待办事项管理需求时，确保不会错过重要的项目。示例如下：

- 在验收标准的应用过程及与需求利益相关方开放的沟通过程中，确保史诗和用户故事清晰明确。敏捷要求透明化，而 CMMI 强化了这一要求，特别是在需求方面。
- 强化这一理念：工作执行团队所做的承诺是产品开发中至关重要的一部分。CMMI 阐明秉承此理念对于需求变更也很重要，这一点经常被遗漏。
- 设定期望：不管采取何种形式，必须确保需求的可追溯性。在敏捷项目中，实现良好的可追溯性可能很难，尤其是在使用纯粹的视觉管理方法时。

当反馈环路较短且团队成员不断进行经验教训总结时，敏捷团队才会成长。“需求开发”的说明性资料中提供了关于如何确保产品待办事项列表中的史诗和用户故事完整可靠的绝佳方法。

CMMI 主任评估师的观点：史诗

史诗是作为需求获取和开发过程结果的工作产品，因此与 RD SP1.1 和 RD SP1.2 密切相关。史诗可在会议室模拟 (CRP)、头脑风暴研讨会或通过其他需求获取过程创建。产品所有者也可以为敏捷团队提供整套用户故事，而不是单个史诗。在这种情况下，相关实践的表述将有不同的上下文环境，并将在“用户故事”部分描述。在另一可靠场景中，可能会在史诗和用户故事开发中执行 RD SP1.1 和 RD SP1.2。其冗余小，可以反映敏捷史诗和用户故事开发的迭代式和增量式特性。

因为史诗必须映射至其下属的用户故事，所以这构成第一级需求可追溯性 … 史诗到相关故事的需求。因此，REQM SP1.4 直接映射至此活动。此外，当你将定义和了解需求视为这项工作的重要一环时，所有需求管理技术可能都适用。

需求开发 (RD)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
获取需求	此实践确保在产品生命周期的早期（例如创建产品路线图期间、发布策划期间，从组建焦点小组、团队的时候），与所有关键利益相关方合作撰写史诗。	产品所有者将高级特性或需求（“要求”）以及期望提供给敏捷团队，这些可分解为用户故事。（参见“待办事项梳理”）	Scrum 看板上的用户故事卡；相应工具（SharePoint、Jura 等）中的用户故事
将利益相关方需求转换为客户需求	史诗通常用于代表业务需求，此实践支持由传统需求规范向创建可加载到工具中或置于团队墙上以供所有人查看的史诗与故事的转变。	Scrum 团队需要在待办事项梳理期间，将史诗分解成多个小的用户故事	Scrum 看板或待办事项列表上的用户故事卡；相应工具（SharePoint、Jura 等）中的用户故事

需求管理 (REQM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
了解需求	此实践侧重于确保向相关人员清晰地陈述所需的业务需求和功能，以便能创建用户故事	产品所有者开发史诗并向 Scrum 团队进行阐述	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
获取对需求的承诺	此实践侧重于确保相关人员清晰地了解所需的业务需求和功能，为创建对业务有意义并且可由团队实现的用户故事提供支持	讨论史诗，直到团队清晰了解史诗并认同需求（史诗）	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
管理需求变更	此实践提供与管理史诗待办事项相关的期望，确保遵循过程来评估提出的变更并始终跟踪这些变更。这有助于在产品所有者不断调整工作期望而不考虑对团队影响的情况下减少团队的混乱局面。	产品所有者在产品待办事项列表中修改、添加和删除史诗，并与 Scrum 团队讨论变更	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
保持需求的双向可追溯性	此实践强调通过用户故事 / 测试了解如何实现史诗的价值，并帮助团队确保在部署之前充分处理了适当的史诗	通过可确保产品所有者自信地说明用户故事集将符合为史诗定义的验收标准的方式，将史诗追溯到高级别的史诗和 / 或相关的用户故事。 原始项目之间在产品待办事项列表、史诗、用户故事以及被确定为“完成的标准”一部分的测试方面有清晰的可追溯性。	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；映射到 Scrum 看板上或某个工具中的相关史诗和高级别用户故事的用户故事；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
使用集成计划管理项目	史诗影响了交付整体产品路线图以及待交付产品的优先级 / 时间。它们是用于确定工作范围的最高级别需求。	对史诗进行分解可能需要对其他工作进行分类。应该在待办事项列表梳理期间将其添加到产品待办事项列表中，以进行优先级排序或梳理。	Scrum 看板上或待办事项列表中的用户故事；适当工具中的用户故事

团队估算游戏与策划扑克

概述

团队估算游戏（有时也称为“斐波那契游戏”）是一项敏捷估算技术，它使用故事点和数量级粗略估算方法，估算相对规模。

策划扑克与之相似，是使用游戏卡片来描绘故事点。

团队估算游戏使用一套卡片或便利贴，以及粘贴在看板上的用户故事。在该游戏中，敏捷团队要进行三轮协作性的商谈来确定规模。每一轮中，团队成员可以在看板上更改某一个用户故事的相对规模。每一轮结束后，团队讨论看板上的故事顺序。确定相对规模后，将再次通过三个轮次重新整理看板，以便根据敏捷团队所有成员的意见，将故事点分配给每个用户故事。团队估算游戏使用斐波那契数列作为确定相对规模的基础。

使用 CMMI 强化团队估算游戏

很多项目都无法得出良好的估算结果。敏捷方法试图使用估算游戏来确定待办事项之间的相对规模，从而解决估算问题。待办事项的相对规模故意与以小时数表示的工作量区分开，以迫使团队用不同的方式思考问题。“大致的正确”而不是“精确的错误”是团队估算游戏的最终目标，

CMMI 的“项目策划”过程域有助于进一步说明估算的清晰度和可靠性。

以下是一些示例：

- 强化需求，确保尽早了解完整的项目范围。需要非常详细的信息，以便为每个项目（无论是史诗还是用户故事）确定相对规模，最终达到完整性的目标。
- 设定在估算中需要考虑所有类型工作产品的期望。软件只是工作产品中的一种；还有许多其他工作产品。
- 确保工作量估算是策划的最终结果。工作量可让敏捷团队衡量其速度并改进未来的估算。

“项目策划”过程域的说明性资料中还提供了许多其他示例和建议。

在团队估算游戏中，Scrum 团队与 Scrum 教练和产品所有者合作，确定每个史诗或用户故事的规模，作为产品待办事项列表开发或冲刺策划的一部分。

在团队估算游戏中，具有不同技术背景的团队成員将深入讨论每个用户故事的功能。

清晰的史诗和用户故事可以最大程度地减少产品所有者花时间阐明产品待办事项，并提高团队在冲刺期间“第一次就做对”的能力。那么，CMMI 可以怎样提供帮助呢？“需求管理”过程域中的 SP1.1 提供了一些指导意见。例如：

- 应清晰地定义需求来源，以最大程度地减少困惑和可能的范围蔓延。
- 团队应文档化良好需求的定义，并加以理解和使用。
- 在创建和评审需求期间，沟通渠道应开放、透明。

CMMI 主任评估师的观点：团队估算游戏与策划扑克

团队估算游戏主要反映项目策划 SP1.1 在产品待办事项列表情况下的使用；但是，如果用户故事通过任务和完整的“完成的标准”得到充实，团队估算游戏还可以验证项目策划 SP1.2 和 SP1.4。因为团队估算游戏是一种协作性的活动，建议评估团队参加一次工作会议，以观察游戏是如何开展的。

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算项目范围	此实践确保在项目生命周期中的早期阶段通过史诗和用户故事明确项目的完整范围。确保项目范围完整比进行详细估算更重要。	用户故事是充分定义的，并在会议期间通过多种技术展示	观察斐波那契游戏；在白板上粘贴便利贴或卡片表示任务，并排序；在用户故事卡片上记录点数
估算工作产品	此实践确保阐明项目生成的所有产出物。有明确的期望。	充分确定了“完成的标准”，以便有足够的信息来估算工作量	观察正在开展的游戏；点数记录在用户故事卡片上；DOD 将斐波那契游戏描述为一种估算方法
估算工作量	此实践确保冲刺策划涵盖在完成用户故事之前必须完成的所有活动。	工作量估算完成后，即可根据所需的资源技能水平和所需投入的工作量来估算成本。	更新的项目成本估算

需求管理 (REQM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
了解需求	此实践确保对创建或阐明史诗或用户故事的活动应用规则。这方面的规则可以优化和提高待办事项列表开发与冲刺策划的效率。	产品所有者和 Scrum 团队在团队估算游戏三个轮次中的其中一轮阐明需求	更新的用户故事；按优先级或复杂性排序的用户故事

结对编程

概述

结对编程是一种工程技术，由两名软件开发人员合作完成编码任务。他们通常在同一个工作站工作：一个程序员为“驾驶员”，另一个为“导航员”，后者在前者输入代码时评审代码。虽然这通常导致编程的初始成本增加，但提高代码质量的回报更高。部分情况下，一名开发人员撰写代码，另外一名开发人员执行单元测试。其他情况下，一名开发人员撰写代码，另外一名开发人员坐在他旁边，帮助解释需求并确保提高代码质量。

结对编程有多种好处，包括：

- 让一名开发人员开发单元测试用例，并使用这些测试用例来实时“测试”另外一名开发人员撰写的代码
- 通过重构可提高代码的质量和可维护性

- 提高对代码和架构的理解，因为结对的开发人员将会时常进行交流
- 让一名开发人员担任“导航员”，协助另外一名开发人员撰写高效的代码
- 实时进行代码评审，协助确保符合编码标准和实践

使用 CMMI 强化结对编程 / 验证

结对编程基于“三个臭皮匠，顶个诸葛亮”的理念。敏捷方法行之有效的一个原因是由于许多实践都具有较短的反馈环路。结对编程只是可以应用的众多实践中的一种。无论是如何结对，合作编程中直接而及时的反馈环路都能让验证变得简单而高效。并且，CMMI 可通过扩展验证概念提高结对实践的清晰度和可靠性。例如：

- 工作产品可能不仅仅是结对开发的代码。用户故事经常会产生其他工作产品，例如培训材料、规格说明或用户使用说明。
- 为了实现高效冲刺，应该确定并提供同行评审的输入信息，例如用于报告缺陷的工具和方法。结对编程中的每个人都应该接受培训，并且能执行评审。
- 完成计划的同行评审对于交付高质量的产品是至关重要的成功因素。不应跳过同行评审。
- 应总结同行评审中的经验教训，并将其馈送到冲刺回顾中。例如，运行效果良好的工具、运行效果欠佳的工具、缺失的过程步骤，或者与同行评审任何输入信息有关的问题等等。

“验证”过程域的说明性资料中还提供了许多其他示例和建议。

CMMI 主任评估师的观点：结对编程

结对编程旨在提高与需求无关的代码质量，而且有助于确保代码符合预期的需求，因此，虽然它并不是在每个过程域中都完整描述，但它在一定程度上与确认（VAL）和验证（VER）相关。但是，结对编程注重代码质量这一事实表明了技术解决方案（TS）也是可以为结对程序员提供帮助的另一个过程域。因为这种技术具有协作性和迭代性，所以技术解决方案 GP3.2 也包含在其中。

验证（VER）

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
为同行评审制定计划	此实践确保已考虑工作方法而不仅仅是撰写代码。结对编程可使其他工作产品受益。此实践还确保为结对程序员提供所有的支持方法和工具。	“驾驶员”和“导航员”通常一起处理代码，一起工作。他们都评审过需求并且理解需求和编码原则	需求、用户故事、团队章程和结对编程过程
开展同行评审	此实践确保在结对编程期间开展同行评审	“驾驶员”和“导航员”通常一起处理代码，他们遵循结构化的结对编程过程	代码注释；缺陷报告
分析同行评审结果	此实践确保获取同行评审的数据，并使用这些数据来改进未来的评审。经验教训总结将反馈到回顾中，以便团队的其他成员学习和改进。	“驾驶员”和“导航员”通常一起处理代码，识别和分析缺陷，并且在冲刺结束时会进行包括对结对编程活动和数据分析的回顾	回顾说明；缺陷报告

使用 CMMI 强化结对编程 / 确认

敏捷方法行之有效的一个原因是在于许多实践都具有较短的反馈环路。结对编程仅仅是可应用的众多实践中的一种，结对编程的理念是“三个臭皮匠，顶个诸葛亮”。那么，怎样才能利用 CMMI 的“确认”过程域来使结对编程更可靠呢？答案存在于上述多种确认实践中。以下是一些示例：

- 考虑如何对软件进行确认。如果采用结对编程方法，在目标环境中开发和验证软件至关重要，一种想法是可以让两名程序员中的一人代表最终用户。

- 设定期望，即在结对开发软件期间所有所需设备均已识别并且可用。
 - 确保结对的每个程序员都接受过关于如何使用特定方法和工具来执行软件确认的培训。结对编程可以将经验丰富的程序员掌握的知识高效地传授给经验不足的程序员。
- “确认”过程域的说明性资料中还提供了许多其他示例和建议。

确认 (VAL)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
选择要确认的产品	此实践设定关于要确认什么的期望。结对工作最常应用于软件代码。	结对编程是选定要执行的过程，所有代码都要使用结对编程方法来撰写，这种技术还用于提高代码质量、执行单元测试以及制订相关标准	标准过程将结对编程定义为标准
策划确认环境	此实践设定了期望，即在已计划的确认活动里所需的所有事情都已识别且可用。	为结对编程装配的工作台有两把椅子、适当的办公桌以及多台监视器。在虚拟环境中，可使用屏幕共享。	现场检查结对编程环境；PPQA 报告也可汇报结对编程环境；培训表明环境可用。
策划确认方法	此实践设定了期望，即为团队选择并提供确认方法。结对的程序员必须接受必要的培训。	选择 Scrum 或 XP 作为标准，并且在组织方针 / 期望中定义结对编程	培训材料或方针材料；WIKI 或过程资产库

使用 CMMI 强化结对编程 / 技术解决方案

使用敏捷开发的组织必须通过可靠的技术实践来交付高质量的产品。敏捷并不意味着混乱无序。良好的“完成的标准”至少应包含适用于所有用户故事的通用标准。以冲刺来开展工作的的好处是，在冲刺期间符合通用的“完成的标准”准则，并且不会拖延到较长的开发阶段结束时：事后通常意味着永不。CMMI 的“技术解决方案”过程域需要在开发产品过程中使用可靠的技术决策和实践。以下是一些示例：

- 应确定编程方法；例如面向对象的开发、测试驱动开发或自动生成代码，等等
- 复用策略应明确

- 适用于开发的编码标准
- 必须满足任何代码复杂度的度量元
- 是否要执行单元测试以及如何执行单元测试
- 静态或动态代码分析工具的使用以及处理相关警告的需求
- 如何报告和管理缺陷

共性实践 3.2 也可以通过将满足完成标准过程中发现的改进措施提交到冲刺回顾会议上并最终分享给组外人员这一活动，帮助结对编程过程的可靠性提升。敏捷是制度化的学习，CMMI 在这里也很适用。

“技术解决方案”过程域的说明性资料中还提供了许多其他示例和建议。

技术解决方案 (TS)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
实现设计	此实践确保确定和实现所有计划的设计目标。	“导航员”的任务是确保遵循编码标准（复杂性、命名习惯、变量命名、内部应用程序接口）	代码；代码注释；缺陷报告
收集过程相关经验以便将来改进	此实践确保落实了收集优秀实践和改进措施并将其应用到未来工作中的机制。	“导航员”提供关于代码和过程的改进措施	经验教训；每日站会上的评论；每日站会或回顾中的“当周改进措施”



产品待办事项列表

概述

产品待办事项列表是产品所涵盖的一切事项的优先级列表。它包含功能特性、缺陷错误（如果产品处于正式生产阶段或在之前的冲刺中发布过产品）、文档变更以及产品所有者要求的任何其他任务。产品所有者拥有待办事项列表、待办事项中的任务以及待办事项列表相关任务的优先级。所有者可参考利益相关方、业务代表和 Scrum 团队的意见来设定优先级，但所有者要对此负责。待办事项列表在冲刺期间通过待办事项梳理过程进行维护。

使用 CMMI 强化产品待办事项列表 / 需求管理

产品待办事项列表是产品中可能需要的一切事项的有序列表，也是对产品进行需求变更的单一来源。无论使用什么敏捷实践，产品待办事项列表都是敏捷项目的基础：待办事项越完善，项目结果越好。CMMI 中的“需求管理”过程

域提供许多最佳实践，可帮助敏捷团队确保在创建和改进产品待办事项列表时不缺失重要项目。以下是一些示例：

- 在验收标准的应用过程及与需求利益相关方开放的沟通过程中，确保史诗和用户故事清晰明确。敏捷要求透明化，而 CMMI 强化了这一要求，特别是在需求方面。
- 强化这一理念：工作执行团队所做的承诺是产品开发中至关重要的一部分。这种承诺最终在冲刺策划会议期间做出，并通过高质量的待办事项得到加强。
- 随着时间推移，强化管理产品待办事项变更的需求。CMMI 强调，需要让利益相关方参与变更分析和决策，这也是容易忽视的部分。

“需求管理”的说明性资料中提供了关于如何确保产品待办事项可靠性的绝佳方法。

CMMI 主任评估师的观点：产品待办事项列表

建立产品待办事项列表的内容、用户故事、初步估算和优先级可表示产品所有者了解并承诺产品需求。此数据也为项目计划的制定和相关计划的集成奠定基础。因为产品待办事项列表需要在待办事项梳理过程中进行修改，所以它在项目的整个生命周期内是一个动态文档。

需求管理 (REQM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
了解需求	此实践侧重于确保向相关人员清晰地陈述所需的业务需求和功能，以便能创建用户故事	产品所有者开发史诗和用户故事并向 Scrum 团队进行阐述	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
获取对需求的承诺	此实践侧重于确保相关人员清晰地了解所需的业务需求和功能，为创建对业务有意义并且可由团队实现的史诗和用户故事提供支持	讨论史诗和用户故事，直到团队清晰了解史诗并认同需求（史诗）	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事
管理需求变更	此实践提供与管理史诗和用户故事待办事项的期望，并确保遵循过程来评估提出的变更并始终跟踪这些变更。这有助于在产品所有者不断调整工作期望而不考虑对团队影响的情况下减少团队的混乱局面。	产品所有者在产品待办事项列表中修改、添加和删除史诗和用户故事，并与 Scrum 团队讨论变更	产品待办事项列表中的史诗（在 Scrum 看板上或某个工具中）；参加待办事项梳理会以了解史诗并清楚如何使用它们来开发用户故事

使用 CMMI 强化产品待办事项列表 / 集成项目管理

产品待办事项列表是产品中需要的一切事项的优先级列表。具有大量产品待办事项列表的大型项目可能需要多个团队在同步的冲刺中工作来交付最终产品,协调多个团队的工作提高了项目管理复杂性,可能会对成本、质量和时间安排产生潜在的负面影响。因为像 Scrum 这样的敏捷框架是轻量型的,并且由用户确定实现的详细需求,所以没有对如何管理这种状况提供太多指导意见。幸运的是,CMMI 的“集成项目管理”过程域可以提供帮助。以下是一些示例:

- 确定可能需要在大型项目的产品待办事项列表中记录的其他计划,比如培训、后勤、设施、配置管理以及整合等。

- 需要考虑如何收集一致而有用的改进信息,并在多个敏捷团队中分享。例如,通过冲刺回顾总结的经验教训。
- 待办事项优先级方法对管理多个敏捷团队中的关键路径做出了解释。目标是了解并解释内部依赖关系。

管理复杂项目始终是个挑战和风险来源。“集成项目管理”的说明性资料中提供了关于如何确保高效管理多个敏捷团队的绝佳方法。

集成项目管理 (IPM)

实践	强化敏捷实施的方式:	满足实践要求的条件:	体现方式:
集成计划	此实践为必须在多个敏捷团队之间进行协调(即 Scrum of Scrum)的大型项目产品所有者提供有用的信息	当产品具有相互依赖关系时,产品所有者将与其他产品所有者进行协调	具有优先级的产品待办事项列表设定了与其他项目需求相关联的用户故事

使用 CMMI 强化产品待办事项列表 / 项目策划

产品待办事项列表包含向产品的内部或外部客户交付完整解决方案所需的所有史诗和用户故事。侧重于在每个冲刺结束时交付具有特定功能的软件是对敏捷实践最明显的应用；但交付软件涉及许多事项。所有这些都应该是产品待办事项列表的一部分，以确保估算是完整的，并且项目能兑现承诺。那么，完整的产品待办事项列表应处理哪种类型的内容呢？CMMI “项目策划” 过程域提供了这个问题的答案。以下是一些示例：

- 关键客户或内部项目里程碑。应考虑向客户交付原型以及所需的准出信息。与敏捷团队接头的支持职能部门需要有足够信心，确保理解高层里程碑。

- 需要接受培训的项目成员。可能存在涵盖某些培训详细信息（例如培训方法、时间安排限制和培训师）的产品待办事项。在冲刺期间，经验丰富的敏捷团队成员可能有一个冲刺待办事项，即培训经验不足的团队成员。

制定完整的项目计划是任何项目取得成功的关键因素。“项目策划” 的说明性资料中提供了关于如何确保产品待办事项列表中不缺失关键项目的绝佳方法。

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
制定项目计划	此实践针对可靠的项目计划的内容提供指导意见。产品所有者必须确保产品待办事项列表是完整的，这样团队才能了解时间安排和可交付成果。	产品所有者与 Scrum 团队在发布策划、冲刺策划和待办事项梳理过程中进行协作，为用户故事确定初始和后续的优先级，并以故事点和 / 或工作量的形式提供估算	用户故事按优先级顺序排列并且每个故事附加有工作量的产品待办事项列表

重构

概述

重构技术，通常应用在敏捷环境中，它侧重于改进软件代码的内部结构、风格、运算和条件，同时保持其外部功能和行为。重构往往意味着重写代码、修复缺陷或改进代码值得注意的某些方面或代码的某些功能。重构的意图是通过改进某些属性（长度、重复性和复杂性）来提高代码的可维护性，使代码能被更好地理解，让开发人员能更好地实现设计决策。

重构是改进代码的一种形式，它可以同等程度地关注代码质量、需求质量、测试、集成或开发生命周期内基于过程的任何其他可交付成果。

使用 CMMI 强化重构 / 集成产品管理

敏捷开发的一个好处是通过许多封闭式反馈环路进行学习。反馈是在工作的特定时点上完成的，不会拖延到项目的晚期阶段，那时已没有时间大张旗鼓地学习。精益工程教导我们，以“重”字开头的任何词汇很可能造成浪费。但重构是敏捷团队用于持续改进软件的一种封闭式反馈环路。重构是敏捷开发中的一种良好实践，也可以用来改进其他工作产品。

“集成项目管理”过程域提供可用于扩大重构收益的指导意见。以下是一些示例：

- 识别重构的代码、改进的用户故事或准确的故事规模结果的最佳实践示例
- 除了每一次冲刺回顾之外，设定知识分享的目标，以记录知识并与其他团队创造培训机会
- 将获得的知识提交给负责定义、部署以及为文档化敏捷实践提供支持的支持组织

“集成项目管理”的说明性资料中提供了关于如何在敏捷项目中加强重构的绝佳方法。

CMMI 主任评估师的观点：重构

侧重于持续改进产品的重构与 IPM SP1.7、GP3.2 和 TS SP3.1 相关。

集成项目管理 (IPM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
构成组织过程资产	此实践为应该在整个组织中分享的学习类型提供指导意见。重构是一种潜在的学习来源。	假设记录了通用重构并且在核心团队之外分享	加入到经验教训库中；在培训和指导材料中包含重构结果和示例

使用 CMMI 强化重构 / 技术解决方案

重构是敏捷团队用于持续改进软件的一种封闭式反馈环路。从其他团队获得的知识可应用于多种主题的实践改进。“技术解决方案”过程域提供了指导建议，可用于扩大设计实现时重构的收益。以下是一些示例：

- 在实现代码时使用经过验证的工具配置；例如，静态分析工具、动态分析工具、单元测试套以及仿真工具等等

- 在实现过程中使用重构的代码、改进的用户故事或准确的故事规模结果的最佳实践示例
- 通过培训确保团队了解实现最佳实践

“技术解决方案”的说明性资料中提供了关于如何在产品实现过程中加强重构的绝佳方法。

技术解决方案 (TS)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
实现设计	此实践设定要充分而正确地实现设计的期望。之前通过重构实践学习到的知识，作为实现的一部分。	向 OPD 输入的关于怎样才能实现重构以及可以使用什么方法的信息；作为经验教训提供的文档	假设在核心团队之外分享重构实践和方法

发布燃尽图

概述

“发布燃尽图是按冲刺显示总体发布轨迹的信息辐射器。”敏捷团队可根据故事点的数量在每个冲刺中“燃尽”（也称为“速率”）。发布燃烧图可帮助产品所有者和敏捷团队了解他们是否要按计划在规定的时间范围内交付所需的功能。

原始数据来源于产品待办事项列表。横轴表示团队（在 Scrum of Scrum 环境中是多个团队）的每个冲刺。纵轴表示在每个冲刺开始时剩余的故事点总数。

敏捷团队可以使用燃起图。燃起图基本上与燃尽图是相同的。唯一的区别在于，团队不是跟踪还要完成多少待办事项工作，而是跟踪他们

已经完成了多少待办事项工作，因此曲线的走向是朝上的，而不是朝下的。

敏捷团队成员观点

发布燃烧图是站在较高级别监控迭代发布进度的机制。迭代燃烧图和每日站会跟踪迭代期间的详细活动。

为了在组织中跨多个项目获得协调一致的经验，“组织过程定义 (OPD)”、“组织过程焦点 (OPF)”以及“度量分析 (MA)”等 CMMI 过程域可帮助团队定义适用于不同图表的各种标准模板、在不同项目中使用的指导原则、历史数据以及图表分析指导原则。这些模板总是通过不同的项目经验得到增强。

此外，团队还可以跨项目分享他们的经验，这是 PMC GP 3.2 以及“集成项目管理”带来的收益。

项目监督控制

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
监控项目策划参数	此实践针对发布计划监控发布策划参数的实际值	发布计划是根据发布燃尽图（RBDC）中呈现的数据调整的	更新的发布计划
监控承诺	此实践可确保在发布计划、冲刺计划或其他相关计划中做出的所有承诺均受到监控和跟踪	发布范围是根据 RBDC 中呈现的开发性能与客户一起修订的。 如果 Y 轴上的故事点是按照特性模块或特性组的主题排序和分组的，根据燃烧图数据，该图表将提供那些主题将不按承诺交付。	更新的发布范围 更新的计划工作产品 更新的发布计划
监督利益相关方参与度	此实践确保进行适当的互动，从而提高利益相关方的参与度	在冲刺演示和回顾时与利益相关方一起使用和评审标准 RBDC	发布燃尽图 记录原因和采取的改进行动，以降低发布风险
开展进度评审	此实践期望定期评审项目的进度，了解进度、性能和问题	在冲刺策划会议期间评审并使用 RBDC 提供的数据来调整发布计划和冲刺计划	根据 RBDC 数据更新发布计划和冲刺计划

度量分析

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
详细定义度量元	此实践定义了用于实现度量目标的度量元	使用符合组织目标的可操作的度量定义。基于故事点的速率应作为时间表和预算成功的主要指标	发布燃尽图
获取度量数据	此实践收集实际度量数据	数据在每个冲刺结束时记录并显示，供所有团队成员查看	发布燃尽图
分析度量数据	此实践分析和解释度量数据并生成结果	使用故事点作为主要指标，高层管理者不希望使用较传统的指标（例如，成本绩效指数（CPI）和进度绩效指数（SPI），而不是速率）	发布燃尽图
沟通结果	此实践确保所有度量分析活动都传达给利益相关方	使用 RBDC 作为信息辐射器，并可供所有团队成员查看	发布燃尽图

组织过程定义

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立组织的度量库	此实践侧重于在组织库中建立和维护度量元，以便供项目组获取并使用在后续项目	数据存储于公共区域中，供所有团队成员查看	发布燃尽图

集成项目管理

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立项目已定义过程	此实践确保敏捷团队可以利用公司的标准过程来明确规定特定的团队或项目需要执行什么工作，包括冲刺长度、发布期间、冲刺策划、待办事项梳理、回顾，等等。	发布燃烧图模板、过程和指导原则已记录在公司的标准过程中	公司的一套具有裁剪指导原则和标准的标准过程
使用组织的过程资产来策划项目活动	此实践将以前的策划和执行活动结果用作预测指标，预测所估算工作量的相对范围和风险。发布燃烧图可帮助团队了解以前的性能。	团队使用组织的某个标准模板来制作发布燃烧图；当团队在发布燃烧图中积累了一些数据点时，就可以使用这些点来修订对剩余发布工作和未来产品发布的估算信息	针对发布选择和裁剪发布燃烧图；根据速率历史记录修订估算信息

发布策划

概述

发布策划的目的是承诺制定计划来交付递增的产品价值（产品增量发布）。在这个阶段，团队不会详细计划每项活动，而是推迟详细承诺至迭代策划阶段，例如确定谁打算何时做什么事。

开始进行发布策划之前，需要一份由产品所有者管理的按优先顺序排列的产品待办事项列表。产品所有者负责将产品版本、市场和商业目标传达给团队，并向他们确认发布实现期间产品待办事项列表中是否可能需要新的项目。产品所有者传达对发布的商业期望（固定的发布日期或固定的发布范围）。

在发布策划期间，产品所有者和敏捷团队协作确定将在何时交付哪些特性。大多数情况下是制定一份根据交付时间识别系统功能的“主计划”。敏捷团队可以通过发布计划来策划需要的能力和团队规模，选择迭代长度，并估算速率和期望的冲刺数量，确保在发布截止日期前完成任务。

团队通常为发布增加缓冲迭代来应对来自客户演示的意外变更。但发布计划是动态工作产品。当产品所有者从演示中收到反馈时，可通过与敏捷团队和业务方（客户）协作来修订发布计划。

发布计划往往会确定特性的优先级，这样敏捷团队就可以在早期的冲刺中重点关注价值最高的特性，将价值较低的特性留待后期；如果团队通过进一步了解确定这些特性具有较低的商业价值，甚至还会将其摒弃。

项目策划

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算项目范围	此实践确保已识别的项目任务已足够详细，可以了解项目范围	定义了要在发布期间的某个特定冲刺或一组冲刺中交付的功能	已定义了发布范围的发布计划
确定工作产品和任务属性的估算	此实践的目的是确保在估算项目时使用适当的估算方法。需要有工作产品和任务的属性才能进行估算。	敏捷估算过程和指导原则已记录在公司的标准过程中	组织鼓励根据故事点使用敏捷估算方法；用户故事包含可靠的“就绪定义”（其中包括任务和工作产品）
定义项目生命周期阶段	此实践定义了项目将经历的项目生命周期阶段。在进行项目工作量计划和进度计划时，了解生命周期至关重要。	对于已选择的冲刺长度，必要时可策划“0”迭代和稳定性迭代；发布 / 迭代 / 故事的 DOD 描述发布生命周期；确定了期望的内部演示和客户演示	包含冲刺数量和长度的发布计划；DOD 和计划的演示
估算工作量和成本	此实践根据估算原理为工作产品和任务估算项目的工作量和成本	确定了计划的速率、冲刺数量和资源需求；可以确定项目成本估算值	包含冲刺数量和长度并确定了团队规模的发布计划
制定预算和进度表	此实践制定项目的预算和进度表	确定了发布日期和客户演示日期，发布总规模，为所需的人力 / 非人力资源分配预算	具有发布日期和计划演示日期的发布概要计划；发布规模和预算
识别项目风险	此实践的目的在于项目策划期间识别和分析所有项目风险	为需求的模糊性和不确定性或团队定义的任何其他原因增加迭代作为应急	为应急情况计划缓冲；为峰值定义和计划时间范围；业务反馈记录

项目策划（续）

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
策划利益相关方参与度	此实践提供指导意见来确保识别所有利益相关方并明确其责任。这个概念对使用敏捷开发实践的团队来说也很重要。	假设团队针对发布分配了所需的技能和缓冲，在发布计划中确定利益相关方，并指出他们需要参与哪些敏捷实践（例如，冲刺策划、待办事项梳理、冲刺演示，等等）	发布计划识别了利益相关方和敏捷实践；；具备所需技能和角色的团队配置，团队速率
制定项目计划	此实践针对可靠的项目计划的内容提供指导意见。这包括将项目的所有方面联系在一起。	假设 Scrum 教练、敏捷团队和产品所有者通力合作，确保可以通过预期的团队速率来兑现承诺的发布范围和日期	发布日期、发布总规模和团队速率
获得计划承诺	此实践从负责执行和支持计划实施的利益相关方那里获得承诺。在本例中，这包括敏捷团队和产品所有者。	敏捷团队在产品所有者的参与下制定了发布计划	敏捷团队参与制定发布策划的证据

度量分析

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
详细定义度量元	此实践明确了用于实现度量目标的度量元。所有度量元都应该足够具体，确保精确且可量化。	使用可度量目标定义发布目标，例如可交付成果、缺陷率、代码质量、团队性能，等等	发布目标和相关的度量指标

集成项目管理

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立项目已定义过程	此实践从项目启动至结束贯穿整个项目生命周期建立和维护项目已定义过程	Scrum、冲刺和发布的产品均被定义为组织资产	过程定义；标准周期，或关于冲刺长度和频率的指导意见

需求管理

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
获取对需求的承诺	此实践从项目参与者处获得对需求的承诺	产品所有者和敏捷团队需要对每个冲刺期间所交付的功能（或其他实践所需要的功能）达成一致	明确定义每个冲刺待交付的功能
管理需求变更	此实践管理在项目进行期间当需求演化时而产生的需求变更	发布计划随着冲刺的推进进行迭代式和增量式更新	更新的发布计划
保持需求的双向可追溯性	此实践确保需求和工作产品的双向可追溯性。发布计划包含了需求与冲刺、与代码模块之间的关联关系。	功能被分配到适当的冲刺和代码模块中	发布计划在多个冲刺和代码模块中分配需求



冲刺 / 迭代

概述

在敏捷团队中，完成任务的主要机制就是固定周期的冲刺。冲刺是一个时间箱的事件，团队成员在其中自行认领用户故事以及包括设计、编码、测试和待办事项梳理在内的所有工作。Scrum 团队在冲刺范围内包含的冲刺策划、冲刺演示和冲刺回顾各不相同，但都会在规定的时间箱内完成这些任务。

敏捷团队成员观点

可通过 CMMI 的“组织过程定义”提供的指导意见和帮助来定义冲刺周期以及关于将上述事件放在何处的决策。可借助“度量分析 (MA)”收集和分析的信息收集用于确定发布周期中最佳冲刺数量的数据。

应仔细策划冲刺，以确保实现冲刺承诺，并且在冲刺结束时生成可用的潜在交付产品。CMMI 的“项目策划”和“项目监督控制”过程域针对包含多次发布及多个冲刺的项目策划提供综合指导意见。

CMMI 主任评估师的观点：冲刺 / 迭代

冲刺是用于在敏捷环境中分配工作的主要机制。当与包含多个冲刺的发布进度表相结合时，它们与“项目策划定义项目生命周期阶段 (SP1.3)”密切相关。

评估团队应验证敏捷团队正在兑现冲刺承诺，跟踪开发速率，并且团队没有延长冲刺时长来完成预计的工作。

项目策划

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
定义项目生命周期阶段	协助制定正式且一致的冲刺周期和内容	发布计划定义了冲刺信息，其中包括确定的周期和内容	发布计划，冲刺计划

组织过程定义

实践	强化敏捷实施的方式：	满足实践要求的条件：	您可能看到其他的展示方式：
建立生命周期模型描述	由组织定义冲刺的周期、内容和管理，特别是“scrum-of-scrum”环境中	“组织过程资产”维护冲刺内容和周期的清晰定义	组织过程资产

冲刺燃尽图

概述

冲刺燃尽图是直观描绘冲刺剩余工作量的信息辐射器。冲刺燃尽图帮助敏捷团队了解能否按计划在冲刺结束时交付预期功能。

燃起图基本上与燃尽图是相同的。唯一的区别在于，我们不是跟踪还要完成多少工作，而是跟踪已经完成了多少工作，因此曲线的走向是朝上的，而不是朝下的。

敏捷团队成员观点

冲刺跟踪中使用燃烧图的目的是监控团队工作进度以及管理完成迭代目标的风险，以交付潜在的可用产品。

团队可从 CMMI “度量分析 (MA)” 过程域获益，提高自身的燃烧图分析能力。团队可以使用 MA SP 1.4 和 SP 2.2 来定义和执行度量分析。

应用“集成项目管理 (IPM)”和“组织过程定义 (OPD)”可帮助团队分享问题、制定解决方案和改进措施，这些都是从跟踪迭代燃烧图中获得的经验。

度量分析

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
详细定义度量元	此实践定义了用于实现度量目标的度量元	使用故事点、实际工作量和剩余工作量作为进度表和预算成功的主要指标	冲刺燃尽图
获取度量数据	此实践收集实际度量数据	数据在每个冲刺结束时记录并显示给所有团队成员查看；还用于预测下一个冲刺的速度	冲刺燃尽图
分析度量数据	此实践分析和解释度量数据并生成结果	使用故事点和剩余工作量作为主要指标，高层管理者不希望使用较传统的指标（例如 CPI 和 SPI），而是速率	冲刺燃尽图
沟通结果	此实践确保所有度量分析活动都传达给利益相关方	使用 SBDC 作为信息辐射器，并将其显示给所有团队成员查看	冲刺燃尽图

项目监督控制

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
监督利益相关方参与度	此实践确保进行适当的互动，从而提高利益相关方的参与度	在冲刺演示和回顾时与利益相关方一起使用和评审标准 SBDC	冲刺燃尽图
开展进度评审	此实践期望定期评审项目的进度，了解进度、性能和问题	在冲刺策划会议期间评审并使用 SBDC 提供的数据来调整发布计划和冲刺计划	根据 SBDC 中的数据重新议定的发布计划和冲刺计划

集成项目管理

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立项目已定义过程	此实践确保敏捷团队可以利用公司的标准过程来明确规定特定的团队或项目需要执行什么工作，包括冲刺长度、发布期间、冲刺策划、待办事项梳理、回顾，等等。	冲刺燃烧图模板、过程和指导原则已记录在公司的标准过程中	公司的一套具有裁剪指导原则和标准的标准过程
使用组织的过程资产来策划活动	此实践将以前的策划和执行活动结果用作预测指标，预测所估算工作量的相对范围和风险。发布燃烧图可帮助团队了解以前的性能。	SBDC 中表示的速率提供用于调整未来的策划（待办事项梳理）活动的的数据	冲刺燃尽图 产品待办事项列表

冲刺演示 / 冲刺评审

概述

冲刺演示（冲刺评审）是一种迭代和增量式的协作方法，用于确保所有利益相关方都了解敏捷团队在每个冲刺结束时要交付的价值。

在冲刺评审期间，将评审已完成的工作和尚未完成的计划工作。将向利益相关方展示已完成的工作。这称为冲刺演示。

在每个冲刺结束时、冲刺回顾之前进行冲刺评审演示。众多利益相关方将出席冲刺演示评审活动，也可将该活动视为沟通、验证并确认团队已实现预定目标的一种形式。

使用 CMMI 强化冲刺演示 — 冲刺评审 / 集成项目管理

每个冲刺结束时的目标是生成潜在的可交付产品。对照完成的标准，度量在冲刺中每个用户故事交付的已完成工作。项目利益相关方参与

评审过程，以确保他们的期望得到满足，并且可以在现场解决问题。解释说明也许可以解决问题，否则可能需要调整产品待办事项列表。也可以邀请客户，因为他们是项目中的关键利益相关方。

策划冲刺演示 — 冲刺评审会议并且让所需的利益相关方参与会议是个挑战。如果做得不好，可能会对团队和产品产生负面影响。“集成项目管理”过程域包含可用于强化冲刺演示 — 冲刺评审的实践。以下是一些示例：

- 确保将冲刺演示 — 冲刺评审整合到高层计划的里程碑中。应清晰地识别这些里程碑，并分配至冲刺团队外部的利益相关方的计划中。
 - 及时更新计划，并及时通知利益相关方。
- 冲刺演示 — 冲刺评审是关键里程碑，利益相关方不应对此有困惑。

- 应提前分配冲刺演示 — 冲刺评审的日期、时间和目标，以确保利益相关方做好准备有效参与。在过程中，可在首次会议之前或有新的利益相关方加入时，安排关于敏捷团队工作方式的培训。

CMMI 主任评估师的观点：

冲刺演示 / 冲刺评审

可通过应用“项目监督控制”、“集成项目管理”和“确认”中的实践来改进冲刺评审和演示。

冲刺演示评审是项目里程碑，也是对项目进度的定期评审。

- 关于关键利益相关方出席会议的任何问题需要及时识别及解决，以最大程度地减少业务中断情况。团队应在会议期间解决关于在冲刺过程中所完成工作的任何问题。对于无法解决的问题应该在待办事项列表中添加新项目或更新现有待办事项。团队应在冲刺演示 — 冲刺评审期间学习相关知识，并将信息反馈到冲刺回顾中。

有效的冲刺演示 — 冲刺评审会议是与关键利益相关方建立信任的最佳方式之一。“集成项目管理”的说明性资料中提供了关于如何管理关键利益相关方的绝佳方法。

集成项目管理 (IPM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
集成计划	此实践确保所有利益相关方都了解关键项目里程碑和他们的责任，并识别了敏捷实践。	有产品增量计划、发布计划或类似计划来确定利益相关方以及他们参与各种敏捷实践时的职责	发布计划 / 产品增量计划或类似计划
使用集成计划来管理项目	此实践设定期望：在整个项目中使用和维护计划	如同“发布”计划中所记录的，邀请所有利益相关方参与冲刺评审和演示；评审已完成的工作和尚未完成的计划工作	出席冲刺评审 / 演示会议
管理利益相关方参与度	此实践确保计划中确定的利益相关方会参与他们的活动。关键利益相关方会出席敏捷实践。	如同“发布”计划中所记录的，邀请所有的利益相关方参加冲刺评审和演示；受邀的利益相关方出席冲刺评审 / 演示会议。如果不能出席，会议将改期，尽量保证最多人数出席	出席冲刺评审 / 演示会议
解决协调问题	此实践确保发现并解决与利益相关方参与活动相关的所有问题。发现并解决敏捷实践中的所有问题。	邀请利益相关方出席冲刺评审 / 演示会议。关于正在展示的已完成的工作以及尚未完成的计划工作相关问题进行讨论，产品所有者决定如何应对这些问题（保留产品待办事项列表中的项目还是创建新的产品待办事项）	出席冲刺评审 / 演示会议； 冲刺评审 / 演示的备注

使用 CMMI 强化冲刺演示 — 冲刺评审 / 项目监督控制

每个冲刺结束时的目标是生成潜在的可交付产品。对照完成的标准，度量在冲刺中每个用户故事交付的已完成工作。项目利益相关方参与评审过程，以确保他们的期望得到满足，并且可以在现场解决问题。解释说明也许可以解决问题，否则可能需要调整产品待办事项列表。也可以邀请客户，因为他们是项目中的关键利益相关方。CMMI“项目监督控制”过程域有助于确保团队按计划开展工作，从而强化冲刺演示 — 冲刺评审的结果。以下是一些示例：

- 设定期望：团队出席每日站会
- 确保团队成员确实出席每日站会
- 设定期望：在每日站会中明确已完成哪些工作、还需要完成哪些工作，以及是否存在阻碍计划进度的障碍
- 确保邀请利益相关方参与定义的冲刺演示 — 冲刺评审里程碑
- 设定期望：现场解决或通过变更产品待办事项列表解决在站会或冲刺演示 — 冲刺评审期间识别的任何问题

“项目监督控制”过程域的说明性资料中提供了其他有用信息。

使用 CMMI 强化冲刺演示 — 冲刺评审 / 确认

每个冲刺结束时的目标是生成潜在的可交付产品。对照完成的标准，度量在冲刺中每个用户故事交付的已完成工作。项目利益相关方参与评审过程，以确保他们的期望得到满足，并且可以在现场解决问题。但是，就确认而言，在进行冲刺演示 — 冲刺评审时，这对敏捷团队意味着什么？如果演示是由执行工作的团队主导，是否为确认？CMMI“确认”过程域阐明了对确认的期望，可用于加强工作执行团队的可信度。例如：

- 确保选择了与最终用户需求相关的材料进行确认：例如，最终客户可能需要使用手册、安装说明、培训材料和工作软件。
- 确保正确执行了每个项目的确认工作，并且记录了数据和结果。目标是工作透明化。
- 在执行冲刺演示 — 冲刺评审的过程中，让最终用户或最终用户代表参与进来。有些公司会邀请客户，让他们直接与产品交互。
- 向提供支持的高层管理者报告确认的状态。让高层管理者出席冲刺演示 — 冲刺评审会议是在敏捷团队中树立信心的有效方式。

有效的冲刺演示 — 冲刺评审会议是与关键利益相关方（包括客户）建立信任的最佳方式之一。“确认”的说明性资料中提供了绝佳的附加信息。

项目监督控制 (PMC)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
监控承诺	此实践确保团队兑现所做的承诺	评审已完成的工作和尚未完成的计划工作	出席和参与每日站会
监督利益相关方参与度	此实践确保利益相关方按计划参与活动。出席每日站会是主要方法。	所有团队成员都出席每日站会	出席和参与每日站会
开展进度评审	此实践确保团队正在完成计划的工作。出席每日站会是评审进度的主要方法。	评审已完成的工作和尚未完成的计划工作；必要时采取纠正措施	出席和参与每日站会
执行里程碑评审	此实践确保让利益相关方参与进来，以保证发布的产品符合他们的期望	如同“发布”计划中所记录的，邀请所有的利益相关方参加冲刺评审和演示；受邀的利益相关方出席冲刺评审 / 演示会议 — 如果不能出席，会议将改期，尽量保证最多人出席	冲刺演示结果；缺陷报告； 冲刺评审 / 演示会议的备注
管理纠正措施	此实践确保在必要时进行变更，以确保项目保持正轨。添加新的产品待办事项列表或更新现有待办事项。	添加新的产品待办事项列表或更新现有待办事项	冲刺评审 / 演示会议上提出的问题； 更新产品待办事项列表

确认 (VAL)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
选择要确认的产品	此实践确保确认了正确的产品。对交付软件的敏捷团队而言就是确认软件。	对于在产品 / 冲刺待办事项中优先级较高的项目，当这些项目在冲刺中完成时，需要向最终用户或用户代表展示	冲刺演示结果；缺陷报告； 冲刺评审 / 演示会议的备注
执行确认	此实践确保根据计划完成确认。对交付软件的敏捷团队而言就是在目标环境中确认软件。	最终用户或其代表出席并参与了活动	冲刺演示结果
确定利益相关方并让其参与进来	此实践确保负责确认过程的利益相关方参与活动。最终用户应参与确认。	最终用户或其代表出席并参与了活动	冲刺演示结果；缺陷报告
与高层管理者一起评审状态	此实践确保高层管理者了解确认过程的有效性	高层管理者出席并参与活动，描述并讨论与冲刺演示结果相关的过程步骤或问题	冲刺评审 / 演示结果



冲刺策划

概述

冲刺策划会议在每个冲刺开始时举行，敏捷团队和产品所有者（或客户）会商谈要在即将开始的冲刺中交付什么价值。

在冲刺策划会议期间，将制定冲刺待办事项并确定任务以支持所策划的用户故事。团队成员评估他们可以在冲刺期间完成多少工作，并选择（或被分配）待完成的任务。团队在实际冲刺执行期间，可以决定是分配还是选择任务。

应该每 7 至 30 天举行一次冲刺策划会议，具体取决于每个冲刺的周期。并且在冲刺策划会议中只需要确定要在本冲刺中交付的价值即可。

使用 CMMI 强化冲刺策划 / 项目策划

产品待办事项列表是在冲刺策划期间随着时间的推移估算并确定优先级的。冲刺团队成员偶尔会基于自身的知识或经验水平参与完善待办事项列表。当待办事项可用于冲刺策划时，应对其进行清晰的描述并确定好规模，以便在下一个冲刺中完成。团队的速率已知，因此，产品所有者

可最大程度提高已完成工作的价值。冲刺策划的最终目标是制定一份冲刺待办事项，在其中以小时为单位列示各项任务以及每项任务的所有者。那么，CMMI 的“项目策划”过程域如何能使冲刺策划更可靠呢？某些实践是直接对应到冲刺策划活动的，并且提供可使敏捷团队受益的其他信息。以下是一些示例：

- 确保冲刺范围清晰，没有缺失任何内容。团队很容易在策划中忽略非软件工作产品，这些疏忽会导致无法实现冲刺目标。
- 设定期望：需要针对团队在冲刺期间产生的一切结果进行估算。
- 设定期望：团队完成工作所需的工作量。根据故事点和策划游戏确定的相对规模最终必须转换成要完成工作的小时数。每个团队成员都应该估算自己的工作量。
- 确保每个团队成员都有机会在做出承诺之前评审冲刺计划和估算结果。敏捷的一大优势就是个人所有权，CMMI 也强调这方面的重要性。
- 设定期望：应根据休假、培训或可能影响计划的其他外部因素调整计划。

- 确保每个团队成员在工作开始之前对计划做出承诺。敏捷和 CMMI 都强调并强化团队成员做出承诺的必要性。在 Scrum 中，可通过 Scrum 看板或协作工具查看承诺。

一份容易理解且切合实际的良好项目计划由工作执行团队制定，并由每个团队成员做出承诺，这是项目成功的积极影响因素。“项目策划”的说明性资料中提供了绝佳的附加信息。

主任评估师的观点：冲刺策划

在敏捷项目中，冲刺策划频繁发生，可视为迭代策划方式或增量策划方式，或是视为产品待办事项列表的确认。项目策划、确认和需求管理都与冲刺策划相关，它们的实践可用于改进冲刺策划会议。

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算项目范围	此实践期望工作包（故事）有确定的顺序（确定了优先级的待办事项列表），可加以确认（冲刺燃尽和发布燃起）	待策划的“范围”是具有固定周期的冲刺或迭代	冲刺待办事项
确定工作产品和任务属性的估算	此实践期望团队针对下一个冲刺中包含的工作进行估算。应包含所有工作。	根据故事点为冲刺策划速率	在待办事项列表中标注为每个用户故事分配的故事点
估算工作量和成本	此实践确保组织同时估算工作量和成本。在敏捷开发中，工作通常受限于时间和人员，所以成本需要考虑人数和时间。	团队一致同意，他们冲刺期间无需加班即可完成冲刺待办事项中的任务	冲刺待办事项
评审影响项目的计划	此实践确保已考虑所有会影响策划的事项。期望对所有冲刺待办事项进行开放式讨论。	所有团队成员和产品所有者都出席会议并达成共识	冲刺策划会议备注
使工作和资源水平协调一致	此实践确保计划切合实际并且可以实现。假设仅在计划切实可行时才能做出承诺。	所有团队成员和产品所有者都出席会议并达成共识；针对冲刺待办事项达成共识时考虑团队成员的可用性（例如，某个团队成员可能在休假）	冲刺待办事项
获得计划承诺	此实践期望每个团队成员对分配到的任务达成共识。在敏捷开发中，会在故事和相关冲刺任务上按个人姓名显示承诺。	所有团队成员和产品所有者都出席会议并达成共识	已完成的冲刺计划

使用 CMMI 强化冲刺策划 / 组织过程定义

当冲刺策划完成时，应清晰描述每个待办事项并确定好规模，以便在下一个冲刺中完成该事项。团队的速率已知，因此，产品所有者可最大程度提高已完成工作的价值。冲刺策划的最终目标是制定一份冲刺待办事项，在其中以小时为单位列示各项任务以及承诺完成各项任务的所有者。CMMI “组织过程定义” 过程域为组织提供指导意见，明确当完成敏捷项目冲刺策划时，需要如何描述。但是否有必要创建裁剪的原则、标准、过程和工作说明呢？答案既是肯定的，又是否定的。有必要，应该获取知识并将其提供给要使用敏捷方法来策划工作的团队。没必要，无需使用各种类型的详细文档来描述每一个细枝末节。以下是所需信息的例子：

- 裁剪说明可帮助项目决定是否使用敏捷策划方法。并非所有类型的开发项目都可以从采用敏捷策划方法中获益。提供指导意见以做出最佳决策比不提供任何指导意见更好。

- 关于如何实施敏捷策划方法的信息。这可以通过正式的培训方法来完成，也可以通过在冲刺策划会议期间让缺乏经验的人与经验丰富的人结对来完成。
- 关于如何配置现有工具来为使用敏捷策划方法的团队提供支持的信息。
- 简单的检查单可帮助缺乏经验的团队确保在最初的几次冲刺策划期间没有遗漏任何事项。

敏捷主要是关乎学习和分享知识。“组织过程定义”的说明性资料中包含组织随着时间的推移用于改进工作方式的最佳实践。

使用 CMMI 强化冲刺策划 / 需求管理

在冲刺策划中，团队与产品所有者合作，尝试估算用户故事的规模以创建冲刺待办事项。用户故事是将用户需求（包括要求）转换成从用户的角度描述结果的语言。用户这个词有非常广义的解释，可以是最终客户、组织的客户以及下一阶段的客户，等等。进行冲刺策划时，必须了解需求。清晰的用户故事可以最大程度地减少产品所有者花时间阐明冲刺待办事项的需求，并提高团队对需求做出承诺的能力，确保在冲刺期间计划的工作与需求协调一致。“需求管理”过程域提供了一些关于强化冲刺策划的指导意见。以下是一些示例：

- 通过应用验收标准以及在需求利益相关方之间开放沟通渠道，确保冲刺待办事项中的用户故事清晰明确。敏捷要求透明化，而 CMMI 强化了这一要求，特别是在需求方面。

- 强化一个理念：冲刺期间工作执行团队所做的承诺是产品开发中至关重要的一部分。CMMI 阐明秉承此理念对于需求变更也很重要，这一点经常被遗漏。
- 设定期望：项目工作与冲刺待办事项中的用户故事协调一致。这与承诺理念密切相关，并且鼓励产品所有者在冲刺期间预防变更和范围蔓延。

“需求管理”的说明性资料中提供了绝佳方法，即通过其他方式确保冲刺待办事项可促进高效的冲刺策划。

需求管理 (REQM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
了解需求	此实践确保对创建或阐明史诗或用户故事的活动应用规则。这方面的规则可以优化和提高待办事项列表开发与冲刺策划的效率。	产品所有者出席冲刺策划会议，并协助团队了解用户故事，以便充分地识别任务	出席冲刺策划会议
获取对需求的承诺	此实践确保相关人员清晰地了解业务需求和所需的功能，为创建对业务有意义并且可由团队在冲刺中实现的用户故事提供支持	所有团队成员和产品所有者都出席会议并达成共识	出席冲刺策划会议；已完成的冲刺计划
确保项目工作与需求之间协调一致	此实践发现需求、项目计划与工作产品之间的不一致，确保在冲刺之前解决这些问题	冲刺待办事项中的用户故事从产品待办事项列表中删除，或应产品所有者的请求，经过协商将其重新放回产品待办事项列表中	更新的冲刺或产品待办事项列表

团队协议

概述

团队协议是敏捷团队成员签订的社会契约，用于定义团队的行为、期望和达成一致的标准，有时也称为“工作协议”。

有些团队协议只是写在看板上的简单想法，有些则是包含关于团队本身重要元数据的详细项目章程。

团队协议通常是在一个发布阶段的初期制定的，可在每次冲刺回顾或冲刺演示后重构。

使用 CMMI 强化团队协议 / 集成项目管理

形成自我管理的团队是实施敏捷实践的重要目标，并且自我管理的团队也是在任何组织中取得敏捷工作成功的关键因素。任何人（甚至是 Scrum 教练）都不会告诉开发团队如何将产品待办事项列表转变成增量式潜在可发布的功能。开发团队的每位成员可能有各自擅长的技能和个人所关注的领域，但责任属于整个团队。CMMI 的“集成项目管理”过程域提供一些有助于团队进行自我管理的方法，此信息对使用敏捷开发技术的团队来说很有用。以下是一些示例：

- 创建共同愿景，可增强团队归属感并形成共同目标。开发团队与外部的利益相关方之间的接口需要明确。共同愿景应该是看得见的、有说服力的，且能被利益相关方理解。
- 确保在必要时可提供团队实现共同愿景所需的资源。建立一个简单而有说服力的团队愿景是让高层管理者为项目提供支持的绝佳方法。
- 设定期望：共同愿景应该及时更新并根据团队在项目开发过程中获得的经验教训加以改进。这非常契合敏捷目标，即较短反馈回路和持续学习。

“集成项目管理”过程域的说明性资料中还提供了许多其他示例和建议。

CMMI 主任评估师的观点：团队协议

可通过应用 OPD 和 IPM 中的实践来改进团队协议，如果敏捷团队选择它们来改进团队协议，可从“共性实践 (GP) 3.2”的使用中获益。

集成项目管理 (IPM)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立并维护团队	此实践为创建适用于一个或多个产品开发团队共同愿景提供指导意见	团队协议包含用于定义团队标准和团队信息的详细信息	团队章程 项目 WIKI 信息辐射器（看板、海报，等等）
确定利益相关方并让其参与进来	此实践确保团队包含负责开发的利益相关方	团队协议确定了本地和扩展的团队成员，并确认他们参与开发过程	团队章程 项目 WIKI 信息辐射器（看板、海报，等等）

使用 CMMI 强化团队协议 / 项目策划

形成自我管理的团队是实施敏捷实践的重要目标，并且自我管理的团队也是在任何组织中取得敏捷工作成功的关键因素。开发团队的每位成员可能有各自擅长的技能和个人所关注的领域，但责任属于整个开发团队。创建团队协议时，不仅必须考虑开发团队，而且要考虑影响开发团队或受开发团队影响的所有外部利益相关方。CMMI“项目策划”过程域提供一些方法来帮助确保识别重要的利益相关方并让其参与进来。以下是一些示例：

- 确保将想法应用到确定利益相关方的决策过程中。遗漏关键利益相关方将来会对项目产生负面影响，最大程度地减少这种风险应该是重中之重。

- 确定可能需要向外部利益相关方提供的任何培训，以让他们有效地参与到项目中。支持性职能部门可能不了解团队使用的敏捷实践，这种知识缺口会在组织中产生负面影响。
- 阐明利益相关方何时需要参与。使用敏捷实践并制定了冲刺的团队，其团队成员需要固定。但外部利益相关方可能需要在开发过程中的不同时间参与进来。良好的策划和沟通必不可少。

从传统模式到敏捷模式，对任何类型的开发来说，清晰地了解利益相关方及其责任都是关键原则。“项目策划”过程域的说明性资料中提供了可能对正在实现自我管理目标的团队来说非常有用的信息。

项目策划 (PP)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
规划利益相关方参与度	此实践提供指导意见来确保识别所有利益相关方并明确其责任。这个概念对使用敏捷开发实践的团队来说也很重要。	团队协议确定了本地和扩展的团队成员，并确认他们参与开发过程	团队章程 项目 WIKI 信息辐射器（看板、海报，等等）

技术债务

当敏捷团队在时间、预算或资源的限制下主动决定采用不太理想、不太高效或不太可行的解决方案时，就引发了技术债务。

随着技术债务的增加，继续开发系统或维护现有系统的成本和工作量会变得过高，可在发布中预先安排“技术债务冲刺”以提高代码质量。

技术债务是一种设计方法，它：

- 在短期内是权宜之计
- 长期来看代价很高，因为在这个技术环境下，未来做相同的工作将比现在做花费的时间更长或成本更高

任何开发团队都会面临的挑战是：

- 业务人员喜欢技术债务（短期获益）
- 技术人员讨厌技术债务（长期痛苦）

为了不断改进开发工作，我们必须提高以下方面的能力：

- 量化决策中的技术债务
- 平衡项目的业务需求与技术需求

使用 CMMI 强化技术债务的管理

可使用特定目标 1 来强化引发技术债务的决策：在过程域技术解决方案中选择产品组件解决方案。使用植根于“决策分析与解决 (DAR)”的标准和结构化决策有助于在主动降低代码质量时，作出一致且适当的决策。

敏捷团队还可以利用“项目策划”中的实践，将技术债务作为用户故事分配到未来的冲刺和发布，从而制定技术债务计划、确定技术债务优先级、排定技术债务的消除顺序。

CMMI 主任评估师的观点：

技术债务

主任评估师和评估团队的成员应始终明确，引发技术债务的决策应该是主动做出的，并不是缺陷或编码错误导致的。产品所有者应收集技术债务并分配给用户故事和冲刺，并适当确定优先级。这样一来，它们就代表了项目管理、需求管理和技术解决方案的执行情况。

技术解决方案

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
制定候选解决方案和选择标准	将决策准则和决策结构化应用到关于应选择什么作为候选技术债务的决策中	使用了决策准则，并记录了决策理由（请参阅“决策分析与解决”）	包含影响分析和原理的决策记录

项目策划

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
估算工作量和成本	帮助了解所引发技术债务的影响，以便做出合理决策	进行估算和影响分析	引发技术债务的成本的估算基础
确定项目风险（另请参阅“风险管理”）	帮助团队在执行技术债务之前充分了解组织面临的风险	执行风险分析	可用的风险及缓解计划记录。
获得计划承诺	确保适当的利益相关方承诺承担技术债务的成本	其他得到授权的产品所有者做出承诺	在待办事项列表中记录变更以表示技术债务用户故事

测试驱动开发

概述

测试驱动开发 (TDD) 是极限编程 (XP) 项目中最常用的一种敏捷技术。

作为早期验证和确认的方法，一名开发人员将先编写一个基本的测试用例去验证所开发的功能；然后在知晓该测试将失败的情况下，编写最少量的代码来通过测试。接下来，开发人员将整理代码以确保其满足性能要求以及编码标准和原则（“重构”）

当与迭代方法和增量方法结合使用、并采用敏捷项目中常见的短冲刺或短迭代时，测试驱动开发可带来最大的价值。

使用 CMMI 强化测试驱动开发 / 需求开发

测试驱动开发 (TDD) 可让开发人员以增量方式创建测试用例以及开发符合测试用例的软件。通常使用自动测试框架来确保测试驱动开发实践简单高效。在开发的后期阶段，如果有变更影响到软件，可使用经过验证的测试用例来确保该变更不会造成意外的后果。执行 TDD 这一实践可帮助团队随着时间的推移创建和完善需求，也很契合 CMMI “需求开发”过程域。以下是一些示例：

- 确保考虑操作概念和最终产品的使用场景。许多情况下都存在对用户互动、操作环境、维护以及生命周期结束的需求。操作概念和场景开发是一个迭代的过程，而测试驱动开发就是很好的方法。

- 尽可能识别会在更广泛的产品环境中引起问题或风险的任何需求。时间限制、系统属性和质量属性可能会导致需求冲突、风险增加或成本增加。当使用测试驱动开发时考虑这些因素有助于实现最佳平衡。
- 确保在通过测试驱动开发实践实现需求时考虑目标环境。模拟环境非常适用于开发的早期阶段，但是创建的测试用例应转变为最终产品的需求确认。

“需求开发”的说明性资料中提供了关于如何充分利用测试驱动开发实践的绝佳方法。

CMMI 主任评估师的观点：测试驱动开发

测试驱动开发是可以改善代码质量和需求的强大方法，因此与验证、确认和需求开发有密切的关系。

需求开发 (RD)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
建立操作概念和场景	此实践设定期望：需求应涵盖所有可能的用例。测试驱动开发实践有助于充实需求。	编写足够详细的测试用例，并且在整个过程中随着需求的充实来使用这些测试用例	测试用例
分析需求以实现平衡	此实践设定期望：随着时间的推移详细地完善需求，并使之适用于更大范围的背景下。测试驱动开发实践符合该期望。	测试不断改进，具备足够的深度和广度，可充分涵盖所有方面	经过改进的测试用例
确认需求	此实践设定期望：应由最终用户或最终用户代表确认需求。可使用自动化方法来辅助实施此实践。	测试不断改进，并使用自动化手段确保需求被有效地确认	自动执行的测试用例

使用 CMMI 强化测试驱动开发 / 确认

测试驱动开发 (TDD) 可让开发人员在开发活动开始时以增量方式创建测试用例以及开发符合测试用例的软件。在创建测试用例和开发软件的同时，也在高效的闭合回路过程中创建、更新、更正和完善需求。可以随时随地进行经验教训总结。并且因为需求缺陷是在开发过程中而不是开发阶段后期发现的，自然可以节约成本。但是，怎样才能通过 CMMI “确认” 过程域来强化测试驱动开发呢？如前面多个章节所述，CMMI 包含有用的实践和信息，有助于填补敏捷框架缺少的某些细节内容。确认方面的部分示例包括：

- 确保开发团队在使用测试驱动开发实践创建和完善符合确认需求的测试用例时，考虑了在目标环境中测试产品。
- 设定期望：将验证的结果反馈给开发团队以便其学习并改进需求和确认需求的测试用例。这一期望在许多项目中都容易被忽略。

在开发早期将确认铭记在心并使用测试驱动开发为确认提供支持是避免以较高的成本解决需求缺陷问题的可靠方式。“确认”的说明性资料中包含团队可用于强化测试驱动开发实践的其他信息。

确认 (VAL)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
执行确认	此实践设定期望：由用户或用户代表在目标环境中以理想的方式测试产品。TDD 测试用例库有助于推动此活动。	测试用例具备确认所需的细节和深度	测试用例
分析确认结果	此实践确保已记录确认结果并将其反馈给适当的利益相关方。此期望强化敏捷开发的闭合回路学习目标。	分析测试驱动开发的测试结果，更正代码或用户故事	测试结果

使用 CMMI 强化测试驱动开发 / 验证

测试驱动开发 (TDD) 可让开发人员以增量方式创建测试用例以及开发符合测试用例的软件。执行 TDD 这一实践可帮助团队随着时间的推移创建和完善需求，并直接验证需求。创建测试用例库后，可以很容易地评估需求变更并通过回归测试验证需求变更。使用测试用例库会带来意想不到的效果（帮助用户发现需求中的问题），在初始阶段改正这些问题可以大大节约成本。在引入并交付变更后首次运行产品就取得成功是给高层管理者和外部客户树

立信心的绝佳方法。那么，CMMI 的“验证”过程域怎样才能改进测试驱动开发实践呢？以下是一些示例：

- 设定期望：应在产品开发早期确认并在产品开发过程中经常验证。验证的结果可以大大节约与诊断问题相关的错误排查和返工的成本。
- 设定期望：开发团队使用验证的结果进行学习总结并改进需求和验证需求的测试用例。这一期望在许多项目中都容易被忽略，但它是 CMMI 和敏捷开发的核心原则。

在初始阶段消除缺陷是成功交付项目的可靠方式。“验证”的说明性资料中包含团队可用于强化测试驱动开发实践的其他信息。

验证 (VER)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
执行验证	此实践设定期望：对产品进行测试以确保其符合需求。测试驱动开发实践和测试用例库有助于推动此活动。	可跟踪测试用例、用户故事和完成的标准以确保对所有用户故事进行测试	描绘可追溯性的 Scrum 看板；基于用户故事功能的足够充分的测试用例
分析验证结果	此实践确保记录验证结果并将其反馈给适当的利益相关方。此期望强化敏捷开发的闭合回路学习目标。	分析测试驱动开发的测试结果，更正代码或用户故事	对测试结果的分析



用户故事

概述

用户故事是从用户的角度对离散的功能项所做的简单的语言描述。用户故事通常是从一个较大的“史诗”或用户故事细分下来的，它是可在单个冲刺中交付的独立功能项。

用户故事是对冲刺进行估算的基础，通常以故事点为单位估算，由此得出给定敏捷团队的项目速率。

用户故事最初描绘在单张“卡片”上并置于 Scrum 看板上，也可以描绘在任何适当的需求管理工具中。

用户故事通常为产品所有者（客户或客户代表）所有。

使用 CMMI 强化用户故事 / 需求开发

用户故事是敏捷项目中的需求。用户故事是将用户需求（包括要求）转换成从用户的角度描述结果的语言。用户这个词有非常广义的解释，可以是最终客户、组织的客户以及下一阶段的客户，等等。清晰的用户故事可以最大程度地帮助产品所有者减少阐明产品待办事项需求的时间，

并提高团队在冲刺期间“第一次就正确完成”的能力。“需求开发”过程域中的 CMMI 特定目标和实践可以清晰地阐明创建用户故事的过程，还有助于提高产品待办事项列表的可靠性。以下是一些示例：

- 确保在创建产品待办事项列表时考虑所有可能的产品需求来源，而不仅仅是外部客户的需求。在项目的早期阶段遗漏重要需求会在后期产生巨大的负面影响。
- 设定期望：对用户故事中记录的需求分配既符合业务需求、又符合产品客户需求的优先级。这将纳入到待办事项梳理和冲刺策划中，两者都必须很好地执行。
- 设定期望：以最佳方式分配用户故事进行开发并最大程度地减少外部依赖性。
- 确保考虑接口限制并在用户故事中加入表示。随着项目的推进以及学习到的更多知识，应在必要时更新接口。
- 设定期望：必须有必要而充分的用户故事在冲刺中驱动开发。产品所有者与开发团队在策划会议上的互动应符合此实践的要求。

- 确保用户故事在目标环境中满足最终用户需求。这应该在开发的早期阶段执行并在开发过程中经常执行，以反复确认产品可行性。

当反馈环路较短且团队成员不断进行经验教训总结时，敏捷团队才会成长。“需求开发”的说明性资料中提供了关于如何确保产品待办事项列表中的用户故事完整可靠的绝佳方法。

CMMI 主任评估师的观点：用户故事

在使用测试驱动开发（TDD）并且已有可靠的完成的标准（DOD）的环境中，可将开发用户故事视为需求开发的可靠形式。非正式地使用用户故事（没有测试驱动开发及完成的标准），将直接导致非正式的需求开发。

使用较短的冲刺或迭代（7-30 天）对测试驱动开发确保需求的完整性和充分性至关重要。

需求开发 (RD)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
获取需求	此实践确保在产品生命周期的早期（例如创建产品路线图期间、发布策划期间，从组建焦点小组、团队的时候），与所有关键利益相关方合作撰写史诗和用户故事。	产品所有者或客户代表提供初始史诗或用户故事	故事卡片；工具（例如 Jira）中的产品待办事项列表
将利益相关方需求转换为客户需求	此实践为创建史诗和用户故事提供充分的支持。此实践包含确定待办事项列表的优先级，其中定义了实现的约束条件。	史诗或大型用户故事被细分	故事卡片；工具（例如 Jira）中的产品待办事项列表
分配产品组件需求	此实践确保对需求进行分配，以增量方式交付产品功能，从而为冲刺策划提供依据	将用户故事细分为定义产品功能的任务；为每个用户故事定义完成的标准	故事卡片；工具（例如 Jira）中的产品待办事项列表；任务看板上的任务
确定接口需求	此实践为将接口需求记录为用户故事的需求提供支持	分解用户故事可以确定其他接口需求	故事卡片；工具（例如 Jira）中的产品待办事项列表；任务看板上的任务
分析需求	此实践侧重于确保所有史诗和用户故事都是必要的，并且支持较高级别的需求	采用测试驱动开发来确认需求；以足够详细的程度定义完成的标准	初始测试用例；完成的标准
确认需求	此实践为确保史诗和用户故事符合最终用户需求提供支持。	使用测试驱动开发来改进代码，并且使用较短的迭代	已完成的测试用例；测试用例版本

速率

概述

“速率”是敏捷团队在冲刺和发布期间用于能力策划和跟踪当前工作的工具。速率以达成共识的工作单位（通常是故事点）为基础，它表示对于特定的 Scrum 团队在即将开始的冲刺中，预期交付多少价值或实际交付多少价值。

速率是特定于团队和持续时间的，这意味着在团队发生变化时，我们就无法合理地认为其速率（交付的价值）会保持不变。团队或冲刺持续时间的任何变化都会影响团队按照给定的速率交付价值的能力。

在冲刺策划期间，敏捷团队将为冲刺设定速率目标。使用燃尽图（BDC）或燃起图作为实时描绘团队实际速率的“信息辐射器”。

敏捷团队应该明白，速率仅仅是对组织有价值的众多度量元中的一个例子。指标本身并不足以管理软件项目。

使用 CMMI 强化速率 / 度量分析

敏捷团队的速率（完成工作的能力）是用于在冲刺和发布期间改进能力策划和跟踪当前工作的基础度量元。可使用数据创建很多度量元，有些度量元会提供有价值的信息，有些则不会。冲刺燃尽图和燃起图可以建立在故事点或工作量基础上，但同时使用二者可以更好地体现冲刺的实际进度。应充分思考这些类型的注意事项，还可以使用 CMMI 的“度量分析”过程域确保敏捷团队使用的度量元更加可靠。以下是一些示例：

- 确保团队制定的度量元与企业的一个或多个商业目标一致。该方法可以在投入时间进行收集数据之前剔除不必要的度量元。
- 设定期望：应该指定需要针对度量元收集哪些数据、如何收集数据、何时收集数据，以及明确最终结果对其支持的目标有何意义。

- 确保将结果传达给受影响的所有利益相关方。支持性职能部门和较高级别的经理希望了解这个所谓的“敏捷”是如何帮助他们做得更好的。
- 不断学习和改进工作完成方式，使其成为团队的关键特征。闭合回路学习是敏捷团队的基本原则，定义完善的度量元可强化这种实践。

上面列出的所有示例都是常识，但很多时候，项目和组织会为了制定度量元而制定度量元。度量元越多不一定越好。“度量分析”过程域的说明性资料中包含可使任何度量计划更加可靠的知识。

CMMI 主任评估师的观点：速率

速率是敏捷团队的核心度量元，也是估算和跟踪项目工作的基础。因为敏捷团队是在“固定时间范围”、“固定团队规模”的环境中工作的，所以像 CPI 和 SPI 这样的度量元是多余的，速率才是衡量项目是否成功的重要度量元。

因为速率对特定的敏捷团队来说是唯一的，所以无法在组织级别预测速率，除非所有团队在整个发布期间不发生变化。

度量分析 (MA)

实践	强化敏捷实施的方式：	满足实践要求的条件：	体现方式：
制定度量目标	此实践期望度量元符合业务需求，并提供关于该需求的有用信息	管理层将速率视为确定项目是否成功的核心度量元，在每个冲刺策划会议上设定速率目标。	具有与用户故事关联的速率目标的冲刺计划
详细定义度量元	此实践期望存在描述某个度量元所有重要属性（包括如何在业务环境中分析数据）的信息	存在描述如何记录、分析、展示和沟通速率的规范	项目或组织的 WIKI 方针文档 培训材料
详细定义数据收集和存储步骤	此实践期望存在关于如何收集所需数据以及将数据存储在哪里以供未来参考的信息	建立了规范及信息辐射器	BDC/BUC 之类的信息辐射器，或工具中的可视化展示
详细定义分析步骤	此实践期望存在关于如何使用收集的数据进行计算的信息	为团队建立了燃尽图或其他跟踪系统；使用可提供虚拟 BDC 的工具	可视化 BDC 虚拟工具 状态报告
获取度量数据	此实践期望按规定收集和分析度量数据	使用燃尽图或其他跟踪系统表示实际项目数据；使用可提供虚拟 BDC 的工具	实时描绘速率的任何信息辐射器
沟通结果	此实践期望团队使用结果来不断学习和改进。还应将结果传达给项目的所有利益相关方和较高级别的经理。	使用燃尽图或其他跟踪系统；使用可提供虚拟 BDC 的工具	实时描绘速率的任何信息辐射器

参考资料、白皮书、演示文稿

CMMI or Agile: Why Not Embrace Both! (《CMMI 还是敏捷：为何不同时支持两者！》，白皮书)

<http://cmmiinstitute.com/resources/cmml-or-agile-why-not-embrace-both>

CMMI vs. Scrum.No!CMMI + Scrum! (《CMMI vs Scrum。不！CMMI 加 Scrum ！》，文章)

<https://www.linkedin.com/pulse/20140917030508-798283-cmmi-vs-scrum-no-cmmi-scrum>

The Elements of Scrum (《Scrum 的元素》，书籍)

Chris Sims、Hillary Louise Johnson、Dymaxicon, 2011 年

Agile Project Management: Creating Innovative Products (《敏捷项目管理：创建创新产品》，书籍)

Jim Highsmith、Addison-Wesley Professional, 2009 年

Implementing Scrum and Agile Together (《同时实施 Scrum 和敏捷》，白皮书)

[https://www.scrumalliance.org/community/articles/2011/february/implementing-scrum-\(agile\)-and-cmmi-together](https://www.scrumalliance.org/community/articles/2011/february/implementing-scrum-(agile)-and-cmmi-together)

Ask The CMMI Appraiser: An Agile Lead Appraiser's Take on the CMMI and Agile (《询问 CMMI 评估师：敏捷主任评估师同时采用 CMMI 和敏捷方法》，博客)

<http://askthecmmiappraiser.blogspot.com>

Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement (《将 CMMI 与敏捷开发整合在一起：关于加速提升性能的案例研究与可靠方法》，书籍)

Paul McMahon

Addison-Wesley Professional, 2010 年

Agile Resiliency: How CMMI will make Agile Thrive and Survive (《敏捷恢复能力：CMMI 如何使敏捷开发得以生存并蓬勃发展》，演示文稿)

Jeff Dalton

<http://cmmiinstitute.com/resources/agile-resiliency>

Manifesto for Agile Software Development (《关于敏捷软件开发的声明》)

agilemanifesto.org

敏捷 CMMI 博客

<http://agilecmmi.com>

Agile/Scrum Developments using the CMMI — NASA (《使用 CMMI 的敏捷 /Scrum 开发 — NASA》，演示文稿)

Kent Aaron Johnson

http://www.nasa.gov/ppt/482611main_2010_Tuesday_4_cmmi_Johnson_Kent_Agile%20Scrum%20Development%20Using%20CMMI%20v1.4a.ppt

Adding Practices to Scrum to Achieve your Goals (《将实践添加到 Scrum 中来实现您的目标》，白皮书)：

<http://www.processgroup.com/pgpostapr2013.pdf>

如需获取更多的信息和资源，请访问

<http://cmmiinstitute.com/resources>

开发团队

CMMI 研究所衷心感谢参与《Scrum 和 CMMI 指南：使用 CMMI 提高敏捷性能》编制工作的众多有识之士。

本指南的编制团队包括主要作者和特约作者，以及每天在该领域运用相关知识技能的 CMMI 和敏捷 / Scrum 专家评审团队，感谢以上人员作出的重要贡献。

特别感谢任甲林先生和郭玲小姐在《Scrum 和 CMMI 指南：使用 CMMI 提高敏捷性能》的中文译文校验过程中给予的专业指导。我们非常高兴任先生和郭小姐为本指南无私地提供宝贵经验，同时我们希望本指南的中文译文能够发挥价值，有效地帮助您提升相关能力。

主要作者

- Jeff Dalton

- Ross Timmerman
- Laura Adkins
- Kirk Botula

- Neil Potter
- Dan Torrens
- Prabhakar S.
- Margaret Tanner Glover

参与作者

- John Voss
- Marion Taros
- Winifred Menezes
- Timothy Zeller

评审人员

- Katherine White
- Paul McMahon
- Winifred Menezes
- Jeff Dalton

市场营销

- Chavonne Hoyle
- Katie Tarara

版权所有 2016 CMMI® 研究所。

无担保

CMMI INSTITUTE 的这份材料是“依原样”提供的。CMMI 研究所不对任何问题做任何明示或暗示的担保，包括但不限于对用途的适用性或适销性、专有性或使用此材料造成的后果做出担保。CMMI 研究所不对是否存在专利、商标或版权侵权的问题做任何形式的担保。

除非在任何附表、附件或附录中有明确规定，否则 CMMI 研究所在法律允许的最大范围内明确拒绝对能力成熟度模型集成 (CMMI) 或根据本协议提供给公司的任何材料或服务进行所有明示、暗示或法定的担保，包括对适销性、特定用途的适用性、非侵权行为、商业上的使用以及使用上述能力成熟度模型集成 (CMMI) 及任何其他材料和服务相关的处理或执行过程的所有暗示担保。

在本报告中任何商标并非要以任何方式侵害商标持有人的权利。

《Scrum 和 CMMI 指南：使用 CMMI 提升敏捷性能》是“依原样、在原处”提供的，没有任何形式的担保，CMMI® 研究所特此拒绝所有明示和暗示的担保，包括但不限于对适销性、特定用途的适用性、准确性、权利或非侵权行为的任何暗示担保。您对本文档负全部责任，并且同意，如果因您使用本文档而给 CMMI 研究所招致任何索赔、责任、损害、成本或费用，您将为 CMMI 研究所提供保护和赔偿，使其免受此类事件的伤害。

© 2016 CMMI® 研究所。保留所有权利

CMMI、CMMI 徽标、Data Management Maturity (DMM) 和 SCAMPI 是 CMMI 研究所的注册商标。

CMMI® 研究所 — 能力成熟度模型集成 (CMMI®) 之家

能力成熟度模型集成 (CMMI) 是一种能力和过程改进模型，可以指导组织实现较高的运营绩效。作为该模型背后的组织，CMMI 研究所正在努力依赖于 CMMI 的成功，推进实践状态，加速开发和采用最佳实践，提供新的和改进的解决方案来满足全球企业的新兴需求。该研究所还主导着人员能力成熟度模型和数据管理成熟度模型。

CMMI 研究所支持全球的大小组织采用其解决方案，涉猎众多行业，包括航空航天、金融、健康服务、软件、国防、运输和电信等。

CMMI 研究所通过交付合作伙伴网络对其产品和服务授权；开展培训和认证工作；为会议、研讨会和活动提供赞助；推广过程改进模型和评估方法的益处。

丰富的遗产

CMMI 研究所是帮助专业人士发挥积极技术潜能的全球性非盈利协会 ISACA 家族的成员。

CMMI 是行业、政府和软件工程研究所（联邦资助的研发中心）成员在卡内基梅隆大学 20 多年来不懈努力的结果。