



数字工业软件

敏捷软件需求管理与合规性： 行之有效的实时方法

高层摘要

敏捷开发方法已经在提高项目成功率方面展现出巨大优势，但仍存在一些未知的领域有待探索。例如：在使用敏捷开发方法管理违反行业标准和法规要求的不合规风险时，如何从开始获取软件需求之际实现信息的可追溯性？本文介绍了一种实时方法，并讨论了敏捷开发方法的采用和改进如何能够显著降低项目失败风险，同时提高合规效率。

目录

序言	3
敏捷原则的价值主张	5
最常用的敏捷开发方法	6
时刻保持敏捷：是真的吗？	7
敏捷开发难题：需求和管控与开发的比较	8
软件开发工具：走在前沿	9
“不堪回首的往日”：离散的工具和数据	9
当下最先进的工具主张	9
全新 ALM	10
将敏捷开发与 ALM 集成	11
实时方法	12
实时方法准则	12
准则 1：单一上级	12
准则 2：单一来源	12
准则 3：单一存储库	12
准则 4：自定义工作项类专用化	12
准则 5：实时功能	13
准则 6：公开	13
实时级别	14
实时信息	15
信息可用性	16
实时方法与敏捷开发	17
结语	18
参考资料	19

I 序言

敏捷原则的演进是为了解决瀑布式开发的局限性，后者主要问题在于直到最后才显示结果、相关方参与过晚，以及存在不必要的测试延迟。如今，敏捷软件开发方法已经成为主流，因为它专注于深入了解并满足客户的需求。

传统瀑布式模型的优势通常可以概括为基于明确规划流程的计划驱动型模型；其中，明确的需求、需求可追溯性和可测试性以及明确的验收标准至关重要。这些方法的优势在于标准化流程所带来的可比较性和可重复性。瀑布式开发通常被认为是风险最小的开发模式，这使其在大型或长期软件开发项目中颇受欢迎，在项目或产品涉及生命危险或自由风险而受政府监控的行业中尤为如此。

但在现实中，没有任何一家企业是只奉行一种规范方法的纯粹主义者。相反，我们是“混合主义者”，我们会将瀑布式开发、敏捷开发（Scrum、极限编程）、统一软件开发过程（RUP）、螺旋模型或其他方法结合起来以满足自己的需求，借此推动项目和企业走向成功。

为了融入这些混合方法，企业正在逐步淘汰容易造成低效孤岛和管理死角的传统工具，取而代之的是应用程序生命周期管理（ALM）工具，这些工具采用统一设计、基于网络且易于定制，能够为多个不断演进的流程提供支持。

本文介绍了一些应用最为广泛的最佳实践（尤其是敏捷开发方法的采用和改进）如何显著降低软件开发风险，实现合规性与项目成功率的双重提升。我们将这些准则称为“实时方法”，因为它们基于敏捷 - 瀑布式混合原则，并使用由现代化 ALM 解决方案提供的实时数据。

事实证明，要通过敏捷开发方法获得卓越表现，研发人员必须能够在激励性的环境中彼此协作，同时最大限度地减少与进度报告、管理层讨论、文档制作等相关的干扰。

例如，考虑使用收集极限编程 (XP) 需求 (“用户案例”) 的方法。客户 (或用户) 应该成为开发团队不可或缺的一部分，实时回答开发人员的问题，而不是作为外部对象。然而，这在现实中很难实现，因为客户通常是一家拥有数千名员工且分布在多个不同国家 / 地区的企业，他们有着错综复杂的需求定义和审批流程。

此外，敏捷开发团队经常开会决定接下来的几天甚至几个小时的目标。但这样短时快速的最佳实践会让经理和高管们感到非常困扰。这些人员需要对项目成本进行长期规划和战略性公司管控。他们需要的是里程碑和可交付成果，而不是对 “今日目标” 的日常评估。

针对项目信息处理的实时方法可以帮助公司协调这些彼此矛盾但同样重要的需求。支持实时方法的工具的引入将为敏捷软件开发的三大关注领域提供支持：公司管控、需求管理和项目管理。任何此类新一代工具和方法均必须将软件需求工程、项目规划和公司管控直接融入到软件研发流程中，在确保研发团队敏捷性的同时，不增加额外的工作或干扰。

实时方法与极限编程、Scrum 或 RUP 之类的方法截然不同。相反，它是一套准则，旨在为软件开发环境和工具确定一个可能的路线图，为其赋予开放性，进而能够以更高的可用性为不同开发方法提供支持，并能够提供关于项目状态的 “实时” 信息。

I 敏捷原则的价值主张

敏捷开发方法已经在提高项目成功率方面展现出巨大优势。以 XP、动态系统开发方法 (DSDM) 和 Scrum 为代表的几种方法得到了相当广泛的应用，这也证明了敏捷软件开发宣言¹背后的大多数原则都受到了客户和开发人员的推崇。

例如，客户非常赞同宣言中的下列陈述：

- “我们最重要的目标是，通过持续不断地及早交付有价值的软件使客户满意。”
- “欣然面对需求变化，即使在开发后期也一样。为了提升客户的竞争优势，敏捷流程掌控一切变化。”
- “经常性地交付可用软件，间隔可以是几星期或一两个月，倾向于采用较短的时间周期。”

另一方面，开发人员非常赞同以下说法：

- “激发个体的斗志，以他们为核心构建项目。提供所需的环境和支持，辅以信任，从而达成目标。”
- “敏捷流程倡导可持续开发。”
- “以简洁为本，它是极力减少不必要工作量的技术基础。”
- “最好的架构、需求和设计源于自我梳理的团队。”

I 最常用的敏捷开发方法

下面，我们简要介绍一些最常用的敏捷开发方法。下述各点引用了与以下讨论最相关的典型实践。更全面的论述可在《敏捷软件开发生态系统》(Agile Software Development Ecosystems)² 中找到。

动态系统开发方法 (DSDM)

DSDM 是快速应用程序开发 (RAD) 实践的扩展和延伸。在所有敏捷开发方法中，DSDM 提供了最为完善的培训和文档。DSDM 有九大基本原则，包括用户必须积极参与、注重频繁交付产品、授权团队进行决策、在整个生命周期内集成测试，以及开发期间的所有更改均可逆等。

极限编程 (XP)

XP 倡导四大价值观：沟通、简单、反馈和勇气。至关重要的是，XP 改变了人们看待变更成本的视角，并主张通过不断重构以及在开发前进行测试来实现技术优势。XP 提供了一个动态实践系统，其作为一个整体的完整性已经得到了证明。除此之外，还有一些其他实践，例如每日站立会议和客户直接参与。

Scrum

Scrum 提供了一个项目管理框架。在该框架中，开发过程以 30 天为一个迭代周期（称为 Sprint），期间需交付一组特定的需求功能。Scrum 的核心实践在于每天召开 15 分钟的团队会议进行协调和整合。Scrum 已经有近十年的历史，并且已经成功地交付了一系列产品。毫无疑问，最广泛采用的敏捷开发方法完全支持敏捷开发的核心价值观：

“我们一直在实践中探寻更好的软件开发方法，身体力行的同时也在帮助他人。由此我们建立了如下价值观：

- 个体和互动优先于流程和工具
- 工作的软件优先于详尽的文档
- 客户合作优先于合同谈判
- 响应变化优先于遵循计划

换句话说，尽管优先级右侧项存在其价值，但我们更重视左侧项的价值。”

I 时刻保持敏捷：是真的吗？

然而，前面提到的敏捷开发价值观并不适用于所有环境。我们以一家跨国公司为例。该公司的软件研发部门跨越三大洲，所研发的产品服务于全球各地的其他部门。

在这种情况下，是否可能与研发团队一起找到“客户”？流程、多语言手册、公司和研发预算、交付周期以及资源规划又该如何处理？

如今，越来越多的大型企业将软件开发活动（以及其他活动）外包给离岸公司和供应商，因此在需求说明和项目进度控制方面的需求也不断增长。在所有这些情况下，代码不是唯一要产生的要素。

这些事实（以及其他诸多因素）推动了能力成熟度模型集成（CMMI）的演进³。CMM 旨在认证软件研发团队的开发能力，而 CMMI 则用于认证公司内部包括软件开发在内的各种业务流程。之所以从 CMM 转向 CMMI，另一个非常有趣的原因是需要将企业流程和合规性与软件开发结合起来。这样看来，CMMI 和敏捷开发方法似乎是彼此矛盾的，因为前者比后者涵盖的范围要广泛得多。

敏捷难题：需求和管控与开发的比较

多年来，Siemens Digital Industries Software 完成了诸多受客户委托的大规模实施项目，帮助客户将敏捷开发团队与企业对提高 CMMI 合规性的需求结合起来。“CMM 视角下的极限编程”（Extreme Programming from a CMM Perspective）⁴也解决了将极限编程和 Scrum 与 CMM 相结合的问题，并且发现 CMM 中的某些领域超出了现有敏捷方法的覆盖范围。这些尚待研究的领域会影响流程和项目控制以及对供应商的监控。

另一个有趣的观点与客户参与有关。对于复杂的系统，通常无法让客户与研发团队坐在一起商讨解决方案，因为正式的需求收集和细化是由数十乃至上百人共同完成的，而这些人往往分散在不同的地理位置。据“敏捷需求、方法和工具” (*Agile Requirements, Methods and Tools*) 13.3⁵ 所述，需求定义活动几乎总是会产生从作者到读者（例如从客户到研发团队）的文档，并且停滞不前（即不支持变更），而这与迭代和变更驱动型敏捷开发方法直接冲突。此外，由于敏捷方法需要团队协作和精深的领域知识，如果承包商或外包合作伙伴在异地工作，便无法实现敏捷开发。

另一个有趣的事实是：“不管大家怎么想，敏捷开发方法并非不涉及项目，只不过它们不像传统技术那样以文档为中心。尽管如此，但对基于CMMI 进行自我评估的企业而言，这仍然是一个问题。”⁶

因此，在大型或分布较广的企业中，软件需求管理、风险管理、合规性、公司管控以及项目管理等方面实际上都可能会因为在研发中引入敏捷开发方法而受到影响。那么，是否存在合理的妥协？

这个问题的答案可以在敏捷软件开发宣言的一个原则中找到：

“提供所需的环境和支持，辅以信任，从而达成目标。”

这可以解释成“为开发人员提供一套完全集成到公司各种流程中的工具集，让他们能够远程协同，就像与客户位于同一位置一样，从而使其工作得到无缝审核和控制。”

实际上，这意味着当开发人员在敏捷场景下使用敏捷工具进行编码时，公司中的其他人必须能够使用自己习惯的方法和工具来定义和细化需求、提交更改、管理测试用例以及跟踪项目状态。

这在现实中是否可行？工具供应商是否提供了满足这一需求的所有工具？

软件开发工具：走在前沿

软件开发及其支持环境和工具正在步入一个历史性的新阶段。这些工具的实际主张遵循一个非常古老的概念，这一概念深深植根于瀑布式软件开发模型中，但有一些例外情况。

“不堪回首的往日”：离散的工具和数据

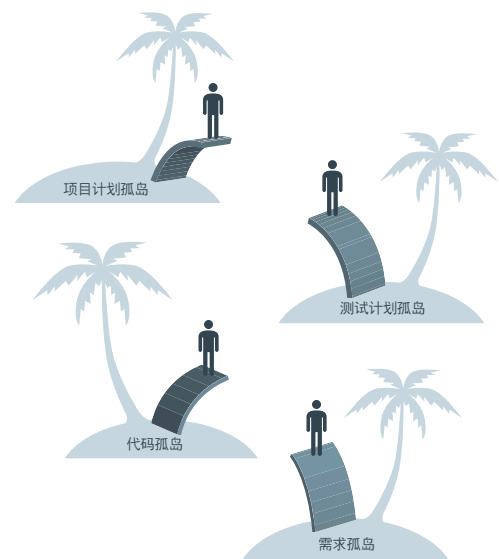
过去，要为软件开发流程的各个阶段提供支持，必须研究离散的工具和数据，这是因为：例如，在某个时间点，要实现协同编程，开发团队显然需要版本管理解决方案。随后，软件架构师及其客户显然需要新的工具，用于需求说明和审批，以便管理复杂的客户 - 供应商关系。诸如此类，不同的阶段需要不同的功能，包括测试管理、更改请求、传播处理以及客户支持等。

相关的问题也会接踵而至：供应商在市场上投放的每一款新工具集都会形成一个新的自动化孤岛。事实上，每一个孤岛都会定义一个新的数据模型、新的访问策略、新的存储库，等等。用户很快就发现，必须对存储在不同逻辑数据模型和存储库中的信息进行集成。于是，供应商们开始搭建连接这些孤岛的桥梁。在很多情况下，为支持某一流程而引入的一些功能被转移到了另一个流程中，以扩展对某些信息集的支持。

完成了这些改进之后，供应商开始向市场提供支持版本化需求、源代码相关更改请求等的解决方案。

当下最先进的工具主张

应用程序生命周期管理是软件开发工具市场中最先进的主张。从前面提到的软件开发工具历史来看，ALM 代表了自动化孤岛集成之旅的最新进展。不管分析师怎么说，传统工具供应商永远无法使用古老代码实现 ALM，因为无论从经济上还是道德上来说，他们都不能抛弃那些支付年度维护费相对缓慢的用户。为了顺应 ALM 趋势，传统供应商试图为其客户提供针对不同供应商提供的工具的一次性集成解决方案。然而，这些解决方案不仅成本高昂，而且并不可靠。这点程度的努力只不过是工具之间的简单数据交换，并不敏捷，当然也不是我们所推荐的实时方法。



全新 ALM

仔细研究全新的 ALM 主张，便会发现以下共同点：ALM 解决方案具有广泛性，能够覆盖各种不同的功能集，支持多个项目角色，并为每个项目角色所需的功能集提供深入全面的支持。

ALM 解决方案几乎总是离不开有趣的准则，甚至是全套方法，以便为软件开发提供支持。当然，方法与工具是集成在一起的。这种（兼具广度和深度的）集成正是客户希望 ALM 解决方案所提供的。

除了集成之外，客户还希望对分散的团队进行公司管控、风险管理、合规管理、全面的项目进度跟踪以及项目相关成本控制。这些都不足为奇。真正的 ALM 系统是统一的，具有统一的链接数据和工作项，用户可以轻松地企业的标准报告中对其进行搜索查询和呈现，而无需依赖于开发人员、IT 人员或承包商。

总之，ALM 解决方案能够有力支持在软件开发中担任不同角色的人员，为各个角色提供深层次的功能集，并辅以集成功能，以此连通由套件中的不同工具形成的信息孤岛。

I 将敏捷开发与 ALM 集成

Gartner 信息技术研究和顾问公司也认为，将敏捷开发与 ALM 集成才能获得最佳结果。在其“ALM 的关键问题” (Key Issues for ALM) 中，他们表示：“无论是部署敏捷开发方法的项目，还是分散在不同地理位置的项目（应用程序由位于世界各地的团队构建和维护），抑或是复杂的流程和产品开发情况，都能够受益于更富成效的 ALM。”借助混合流程，您可以获得所需的灵活性，并实现更好的控制。

尽管此类 ALM 解决方案已经被很多公司广泛采用，但敏捷开发团队并不买账，原因有以下几点：

- 他们认为支持其“集成”方法的集成工具有违敏捷文化。
- 在敏捷开发过程中，所有活动并行进行。而在传统 ALM 工具集成中，需要将信息从一个存储库批量传到另一个存储库（孤岛之间的桥梁），这阻碍了更改的即时通知。
- ALM 解决方案为不同流程中的不同角色提供不同的工具。而在敏捷开发过程中，整个团队在同一环境中协作，使用相同的工具和方法。
- 敏捷开发方法最受青睐的一点在于人员可互换，而 ALM 则更倾向于项目角色专业化。

实时方法可帮助开发人员通过自己习惯（和需要）使用的工具实时访问更广泛的企业信息，从而解决将敏捷开发团队集成到更广泛的公司 / 企业架构中的问题。

我们来总结一下到目前为止讨论的内容：

1. 在软件开发中使用敏捷方法可以取得可圈可点的效果。
2. 在很多情况下，区分敏捷团队颇为困难：他们通常是更广泛的、毫不敏捷的企业流程的一部分。
3. 对敏捷团队而言，仅具有可用的软件开发工具是远远不够的，他们必须参与到更广泛的企业流程中，如风险或合规管理等。

| 实时方法

Siemens Digital Industries Software 提出的实时方法包含一套准则和分类。这些准则引入了一种用于管理软件开发项目和开发相关信息的新理念。从这些准则中还产生了一组定义分类的标准，用于评估实时方法在开发环境和工具中的应用程度。

实时方法准则

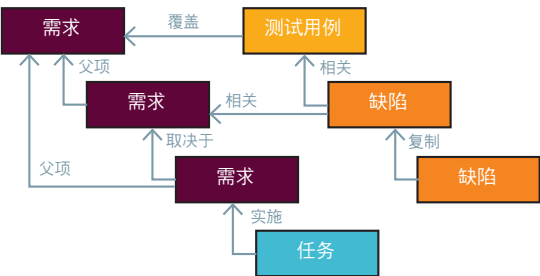
实时方法准则（以下称为“实时准则”）主要分为两大方面：

- 1. 与实时信息、信息模型和存储相关（准则 1 到 4）
- 2. 与信息可用性、实时信息的访问和公开方式相关（准则 5 和 6）

准则 1：单一上级

工作项类是继承层次结构中与管理活动相关的所有信息和项目的共同上级。它的实例被称为工作项。

更为概括地说，根据该准则，在软件生命周期中要创建的所有项目和要执行的所有活动（例如，需求、更改请求、任务、测试计划）都是工作项。也就是说，从需求到代码，在软件开发过程中管理的一切内容都是工作项类的实例。



准则 2：单一来源

工作项类的实例及其所有专用化的实例均为“单一来源”——所有项目信息在开发环境中只存在一次。我们以在需求说明期间创建的测试计划为例。与生成计划的需求相关的测试计划和与执行过程中发现的缺陷相关的测试计划必须为同一计划（而非其副本）。

准则 3：单一存储库

用于存储工作项的存储库在逻辑上应唯一。这并不意味着存储库在物理上必须是一个，但对用户而言，存储库必须唯一。

虽然理论上可以在多个存储库的集成之上构建一个逻辑上唯一的存储库，但是并不推荐这种做法，因为这可能会导致自动化孤岛之间的桥接等情况。此外，还有一些例外情况需要考虑；例如，支持可扩展性需求，使多个存储库能够通过镜像、负载均衡、远程复制等提供更出色的性能。

准则 4：自定义工作项类专用化

用户可以定义自己的工作项类专用化，以满足其公司或项目需求。例如，这可以是通常的“需求”、“更改请求”、“任务”、“源文件”、“代码更改日志”，也可以是“客户采购订单”或任何对组织或单个项目有意义的内容。正是由于这一可能性，用户才可定制存储在工作项类专用化和工作项类本身中的信息。

准则 5：实时功能

当某一功能是适用于工作项类及其专用化的任何实例的操作时，它即为实时功能。换句话说，可以应用于任何工作项的功能即为实时功能。

我们来看一个有助于理解的示例：通常应用于任务以获取任务的实际进度的“显示进度”操作。将此类功能升级为“实时显示进度”，即可使其适用于任何工作项；因此，用户可以获得测试计划、需求说明、更改请求等方面的实际进度。

显而易见，在特定生命周期管理解决方案中，实时功能的数量越多，功能越强大，用户受益越多。

对于垂直工具的提供者来说，创建全新实时功能应该是一项激励性活动：到目前为止，他们一直在为软件生命周期的某个阶段和 / 或开发链中的某个角色改进功能；现在，他们应该思考的是如何扩展这些功能，为软件开发的各个阶段的各种角色创造价值。

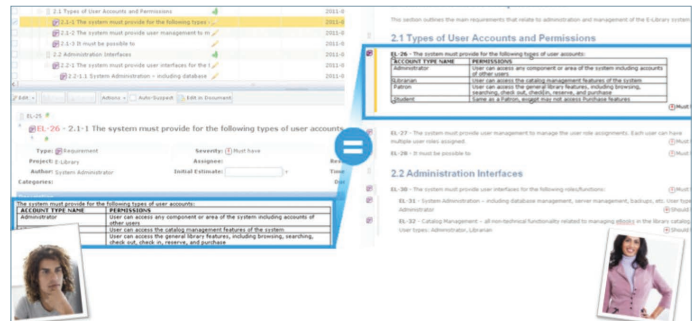
再来看另一个例子：开展实时影响分析的好处。这意味着使用以实际需求为导向的链接导航来查找受更改影响的需求，并对导航进行扩展，以便找到受更改影响的各个开发项目和活动。因此，通过执行实时影响分析操作，用户能够轻松找到为实现需求而执行的所有活动及其成本、相关人员、所有要更改的项目（如源代码和用户手册），以及所实施的更改对特定里程碑的项目计划可交付成果的影响。

准则 6：公开

使用实时功能访问工作项时，所产生的信息应该以适合每个用户角色的方式公开。这意味着不同的用户角色应该能够以所需的格式使用实时功能，并且这些功能可为担任某一角色的用户提供所需的特定内容。

例如，项目经理希望以计划格式访问项目进度信息，执行经理希望在仪表板中只查看达到里程碑的关键路径，而开发人员则希望直接在集成开发环境 (IDE) 中看到分配给他们的任务及其截止日期。

显然，每个开发环境都可以在某个时间点以不同级别满足这些准则的要求。



相同的数据 - 不同的视图。

实时级别

本部分介绍了实时方法的合规分类。分类中包含一组标准，用于检查任何开发环境（即任何类型的开发架构和访问工具集）与实时方法的符合性级别，这种符合性级别在下文中统称为“实时级别”。

分类中包含五个实时级别。最后一个级别完全符合所有准则的要求，并且提供了一组丰富的实时功能，各个不同用户角色能够以所需方式使用这些功能。最后一级是每个工具集在开发工具演进结束时应达到的最终状态，而中间的 1 到 4 级被定义为达到这一状态的过程。这些中间级别有助于公司评估其开发架构对实时方法的实际支持程度，并建议进一步的改进步骤，以进入下一级别。

五个实时级别分别为：

- 1 级 - 基础
- 2 级 - 连接
- 3 级 - 融合
- 4 级 - 控制
- 5 级 - 管控

每个级别都包含用于评估软件开发环境与实时准则的符合性的标准。每一级别均扩展了前一级别的标准。下文讨论了根据领域划分实时准则的标准：实时信息与信息可用性。

管控	实时仪表盘
控制	实时计划
融合	实时跟踪
连接	实时追踪
基础	实时搜索

实时信息

本部分介绍了用于评估软件开发环境与实时准则 1 到 4 的符合性级别的标准。这些标准与信息组织方式相关：数据模型和存储。

1. 基础级别。在这一级别，必须符合准则 1（单一上级）和准则 2（单一来源）。因此，工作项类被定义为公共上级，它的所有实例及其后续实例均为单一来源。
2. 连接级别。工作项通过链接彼此连接。链接必须根据链接角色支持不同的连接类型。例如，工作项可以与“包含”链接或“影响”链接相连接。
3. 融合级别。在这一级别，必须符合准则 3（单一存储库）：所有工作项均存储在单一逻辑存储库中。此外，存储库必须支持针对工作项和链接的版本和历史记录管理，以及至少针对状态更改的工作项工作流管理，包括状态、过渡、被指派者和用户通知机制。最后，存储库必须能够确保安全访问。
4. 控制级别。工作项会存储与时间、优先级和成本相关的信息，并支持讨论和审批。请注意，时间、成本和优先级信息的内容非常广泛，可能包括预计完成时间、计划开始时间、计划结束时间、指定项目里程碑、预期成本、实际成本、价值、附加值、优先级、严重性等。
5. 管控级别。在最后一级，必须符合准则 4（自定义工作项类专用化）。链接类型也由用户定义。因此，工作项类型、内容和连接可以根据用户、项目和公司需求进行自定义。此外，还可以添加风险、资源和财务管理相关信息。

信息可用性

本部分介绍了用于评估软件开发环境与准则 5 和 6 的符合性级别的标准。这些标准与信息的管理方式相关：功能及公开。

要在各个级别添加的实时功能包括：

- 1. 基础级别。实时搜索：工作项可根据各个属性进行搜索。
- 2. 连接级别。实时追踪：工作项支持基于链接角色的导航，以及影响和可追溯性分析。
- 3. 融合级别。实时跟踪：扩展的工作项生命周期管理。
- 4. 控制级别。实时计划：项目规划和进度，其中所有工作项或者属于工作项类的选定专用化的工作项均显示在自更新计划中。这意味着计划是根据存储在工作项中的信息（如优先级、严重性和相关性）自动创建的，并将根据计划工作项的任何更改重新创建。
- 5. 管控级别。实时仪表板：实时管理每个项目活动。仪表板还能够显示多个项目的信息，例如资源负载和跨项目的代码重用情况等。

在各个级别上，实时功能必须以适合用户角色的格式向用户提供信息。在“自动化孤岛”方法中，担任某个角色的用户只能以适当的格式访问特定于角色的信息。在实时方法中，所有用户都能够以所需格式访问所有信息。

例如，以负责项目计划和甘特图的项目负责人为例。为了了解其项目状态，他们必须四处查看包含在 Office 文档中的需求、包含在跟踪器中的问题，以及存储在版本控制系统中的代码等。使用实时方法，他们可以直接以计划格式查看每个工作项的实际状态（即其项目状态）。

实时功能在各个级别提供的信息必须可供每个用户使用。因此，如果实时级别 1 中包含的实时搜索结果可以通过唯一的网络界面提供给每个用户，那么在实时级别 4，不同的用户将以不同的方式处理实时计划：开发人员在其 IDE 中报告自己的进度，项目负责人在甘特图中安排优先级，而管理人员则在电子表格中查看资金成本，并且可以在其中重新分配预算。

级别	实时信息	信息可用性
1 级 - 基础	工作项类被定义为公共上级，它的所有实例及其后续实例均为单一来源。	实时搜索
2 级 - 连接	工作项通过类型链接彼此连接。	实时追踪
3 级 - 融合	工作项包含在支持更改和工作流管理的同一版本化环境中。	实时跟踪
4 级 - 控制	工作项将存储与时间和成本相关的信息。	实时计划
5 级 - 管控	工作项类型、内容和连接皆可自定义。	实时仪表板

表格总结了实时级别合规性的标准。

| 实时方法与敏捷开发

如前一节所述，使用从实时级别 4 开始提供实时和可用项目信息的工具，用户可以轻松应用实时方法来弥合敏捷（在研发团队中）与正式（在研发团队外）流程之间的差距。

例如，在实时级别 4，需求、任务、项目里程碑和项目成本信息，以及更改请求、测试计划、功能、源代码、内部版本等均存在于一个版本化的、完全可跟踪的工作流驱动型存储库中。此外，每个公司或项目角色的所有相关信息都能以角色所需的格式提供。

示例：复杂需求初始阶段

我们举例说明一下。凭借适当的实时方法工具，位于亚特兰大的一家客户公司可以通过 Office 文档在一个环境中执行复杂的需求初始阶段，并进行细化和多级审批；在同一环境中，位于慕尼黑的项目经理能够以甘特图的形式指定任务、优先级和里程碑，而位于德里的开发团队能够以敏捷的方式跟踪其项目和进度。

到达实时级别 5 时，这些分散的团队能够无缝地为位于旧金山的高级管理层提供清晰的信息，从而揭示延迟、瓶颈、成本和风险情况。

Project: E-Library

Categories:

Estimate in Story Points: 100

Time Spent: 3d

Priority: Medium [50.0]

Target Release: Version 1.0

Due Date:

Time Point: 134 (2013-08-27)

Description

User must be able to configure contact information.

Edit

Linked Work Items

Suspect	Role	Title	Project	Revision	Status	
	has parent	EL-110 - 3.1 Basics	E-Library			
	is implemente	☒ EL-50 - Define API to set contact information	E-Library			Steve
	is implemente	☒ EL-51 - Implement IContactInfoService for remote webservice access	E-Library			Steve
	is implemente	☒ EL-52 - Implement: Permissions to set contact info	E-Library			Philip
	is implemente	☒ EL-53 - Implement contact area on user profile dialog	E-Library			Philip
	is implemente	☒ EL-76 - Provide a section on user properties dialog to define contact information	E-Library			Philip

Edit

Work Records

User	Date	Time Spent	Type	Comment
Philip GUI	2011-06-27	1/2d	Analysis	
Steve Developer	2011-06-30	2d	Implementation	

Current

History

Created: 2010-12-02 10:26, Update

结语

敏捷开发方法颠覆了过去几十年中为生产软件而建立的所有方法和工具架构，并定义了一种行之有效的、“自由的”全新工作代码创建方式。不幸的是，由于风险管理、合规管理或更大、更分散的公司的流程导向型环境，敏捷开发方法并不总是行得通。

在保持软件研发部门敏捷性的同时满足企业需求是可以实现的，这要归功于一种全新的软件开发工具，这种工具实际上已经开始在市场上崭露头角，并且在面临以下困境的公司中迅速取得了一

席之地：需要在更广泛的、不太敏捷（或非敏捷）的企业环境中应用成熟的敏捷软件开发方法。由 Siemens Digital Industries Software 推出的 Polarion® ALM 便是这种工具的一个范例，它实际上处于实时级别分类中的第 5 级。

将敏捷软件开发的优势与实时方法中更正式的需求管理、规划和管控方法相结合，能够为未来的研究开辟新的方向，从而创建可满足不同业务部门需求的垂直实时方法和工具。



参考信息

1. *Manifesto for Agile Software Development*, <http://agilemanifesto.org>
2. J. Highsmith, *Agile Software Development Ecosystems*, Pearson Education, 2002
3. CMMI, <http://www.sei.cmu.edu/cmmi/>
4. M. Paulk, "Extreme Programming from a CMM Perspective," *IEEE Software* 18.6, 2001
5. R. Davies, *Agile Requirements, Methods and Tools* 13.3, 2005
6. D. Kane, S. Ornburn, "Agile Development: Weed or Wildflower?" *InformIT*, 31 Aug. 2002

Polarion® ALM™

一体化应用程序生命周期管理解决方案。

关联团队与项目，同时利用集需求、编码、测试和发布于一体的单一一体化解决方案来改善应用程序开发流程。

Polarion® Requirements™

完整的软件需求管理解决方案。

跨整个项目生命周期有效地收集、编写、审批和管理复杂系统的软件需求。

Polarion® QA™

完整的测试和质量管理解决方案。

在单一协同质量保证 (QA) 环境中设计、协调和跟踪所有测试管理活动。

Polarion® ALM	Polarion® QA	Polarion® Requirements	Polarion® Pro™ - 仅任务	Polarion® Reviewer™ – 审核 / 审批
核心功能				
审计、指标与报告				
更改和配置管理				
软件需求管理				
测试和质量管理				
问题和风险管理				
重用与分支管理				
规划和资源管理				
敏捷 / 混合项目管理				
内部版本和发布管理				
变型管理				
PLM-ALM 集成				

完整功能

插件 - 单独许可的功能

Siemens Digital Industries Software

美洲：1 800 498 5351

欧洲、中东及非洲地区：00 800 70002222

亚太地区：001 800 03061910

如需其他地区电话号码，请单击[此处](#)。

关于 Siemens Digital Industries Software

Siemens Digital Industries Software 不断推动数字化企业转型，让工程、制造业和电子设计遇见未来。Xcelerator 是 Siemens Digital Industries Software 推出的软件和服务全面集成式产品组合，可助力各种规模的企业打造数字化双胞胎，带来新的洞察、新的改进机遇和新的自动化水平，让技术创新如虎添翼。如需了解有关 Siemens Digital Industries Software 产品和服务的详细信息，请访问 siemens.com/software 或关注我们的[领英](#)、[推特](#)、[脸书](#)和[照片墙](#)帐号。Siemens Digital Industries Software – 数智今日，同塑未来

作者简介

斯蒂芬·里佐 (Stefano Rizzo) 博士是 Siemens Digital Industries Software 的一名富有远见的产品负责人。他在金融、电信、软件和政府等多个不同业务领域深耕 18 年之久，积累了丰富的 IT 咨询和指导经验。作为一名指导人员，他已帮助数十家大型公司引入了新的开发流程和方法。作为一名培训人员，他为数以千计的员工提供了 UML、需求管理和敏捷开发方面的培训。作为一名方法传播者，他与数百家公司分享了自己关于协同软件开发的愿景。目前，他主要致力于研究和开发敏捷与实时软件开发流程的方法及最佳实践。

siemens.com/software

© 2022 Siemens. 可在[此处](#)查看相关西门子商标列表。其他商标属于其各自持有方。

84383-D2-ZH 1/22 LOC