NUPP

Nearly Universal Principles of Projects

This is a downloadable version of the online manual (https://nupp.guide), generated on 2025-06-02. Please check the website for newer versions.

This manual can be used and distributed freely under the Creative Commons Attribution 4.0 International license.

翻译者: 黄宏

NUPP是英文单词Nearly Universal Principles of Projects的缩写词，顾名思义，NUPP是项目通用原则的集合：无论我们使用什么样的方法论和方法，最好在所有项目中都能够遵循这些原则，以便能够最大化的提高项目的成功率。 项目中的每个人和项目中使用的每个方法，通常只会关注这些项目通用原则（NUPs）的一部分内容，但是：

- 对于从业者来说，考虑全部的这些项目通用原则，而不仅仅是其中的一部分会更有帮助。
- 因为，如果对这些项目基本原则的理解还不够清晰的话，很多从业者往往会深入到实际的细节当中，而忘记了这些原则，并因此做了与这些原则相违背的事情。

项目通用原则（NUPP）兼容所有主要方法、体系、资源和框架，如PRINCE2®，PMBOK® Guide，P3.express，PM², DSDM®，XP和Scrum，虽然它可能与这些体系中的某些解释不一致，实际上，项目通用原则（NUPP）鼓励从业者重新考虑这些解释。 这里介绍的项目通用原则（NUPs）包括：

- NUP1: 结果与事实比隶属关系更重要
- NUP2: 将精力和资源用在最重要的事情上
- NUP3: 积极主动
- NUP4: 细节决定成败
- NUP5: 先定目标，后行动
- NUP6: 使用可重复的元素

NUP1

原则一：结果与事实比隶属关系更重要

我们都有一种自然的倾向，即认为我们隶属于某些群体，这种倾向往往超出了其本身，使我们产生强烈的隶属关系，并因此导致一些问题。 因为这些隶属关系，我们失去的东西比获得东西还要多。 如果我们不将我们的身份和偏好限制在某些群体中的话，我们就可以成为更专业和更成功的专家。

示例：敏捷vs瀑布

在预测型方法作为标准方法的年代，一群高度热情的人在IT开发中勇敢地尝试采用自适应开发方法，并将这种方法称为“敏捷”。这是一个伟大的举措，因为他们并没有将选择限制在看似必要的范围内。在敏捷社区中有许多热情以及结果导向的人，但不幸的是，在这个社区中也有一些人将敏捷作为宗教一样来崇拜，并将所有外人视为敌人。这会在多个方面产生问题，包括：

- 不允许他们向群体外的任何人学习
- 不鼓励群体外人学习他们
- 属于这个群体要比真正的目的更重要，这阻止了许多成员学习敏捷的真正含义

通过将“敏捷”仅作为一个开发方法来使用，而不是将其看作包括很多成员的社区，那么，即使不能解决全部问题，也可以显著的减少这些问题的发生;要使那些认为自己是创造者、问题解决者和领导者的人认识到，他们只是敏捷的推动者之一，而不是敏捷的标志。

对于真正的专业人士来讲，并没有敏捷与瀑布之争。

示例：PRINCE2® vs PMBOK®

社区中有许多专业人士将要么将自己与PRINCE2®关联，要么与PMBOK®指南关联（通常是因为他们的所处的位置），这实际上并不是他们对某一个更加熟悉。 我们都可以对某些资源有所偏爱，但他们不是我们的标志，更重要的是，我们必须熟悉所有这些资源，以拓宽我们的视野和选择。

真正的专业人士对所有想法持开放态度，探索他们，学习他们，并在需要时使用它们，无需隶属关系。

示例：持续学习

隶属关系可以让一个人通过产生归属感而获得满足，但是这种关系并不会推动他们学习，有时甚至因为担心失去他们，而不鼓励他们学习。 当你是一个自由的人、一个没有隶属关系的专家时，你需要通过学习，持续的来学习来填补空白。

我们今天所认为的并不是事实，而是到目前为止我们最好的理解，它需要我们持续改进。 如果一个人的想法与他几年前的想法完全相同，那肯定是出了问题。 甚至对于基项目通用原则（NUPP）来说就是这样，如果你在几年后回过头来，看到完全相同的东西，你肯定会提出质疑。

示例：开放性

当拒绝某个人的时候，请确保你的目的是拒绝那个人的想法，而不是那个人。 这有助于防止很多紧张的情形发生。 同样地，当某人拒绝你或关注你的时候，尽量不要把它当作是针对你的战争，而是在讨论你的想法，并对其保持开放态度。 不要听他是如何回应的，而是要听他是如何理解的; 并与这个人一起提升这个想法。

有些人可能故意针对你这个人，而不是你的想法，在这种情况下，你应该在继续之前来帮助他们关注你的想法，而不是关注你本人，并试着在整个谈话过程中保持这样的状态

NUP2

原则二：将精力和资源用在最重要的事情上

资源是有限的， 对于项目来说可用的资源也是有限的，你做出正确决策的所花的精力也是如此。 你应该将精力和资源用在最重要的事情上，并帮助其他团队成员也这样做。

示例：80/20法则

在项目管理中，通过花费一小部分的努力，可以获得大部分可能的收益。 在大多数情况下，以100%可能收益作为目标来实现，需要付出高昂的代价，并且这样做也是不合理的。 你在做每件事情的时候都需要考虑这个法则，并且鼓励其他人也这样做。

示例：决策疲劳

我们使用有限的精力来完成所有类型的决策，并且还想要把这些事情做好。 如果你耗尽了大量的精力在不必要或不重要的决策上，那么就没有精力来做重要的决策了，这可能会导致糟糕的结果。 这是你应该避免微观管理的原因之一（PRINCE2®提出“例外管理”原则也是这样的道理）。大家对想法的冲突可能是有益的，但与人有关的冲突通常会对项目造成伤害，其中后果之一就是它会耗尽团队成员的精力。 如果你发现这种冲突，应该尽力找到根本原因，并解决它。

示例：关爱自己

你做出的决策和你表达的意愿都会消耗你的精力。 另一方面，通过休息和吃饭

又会充实你的精力。 所以，你应该关爱自己：确保你有足够的睡眠和休息，并且营养充足。 如果你的睡眠习惯不好或有问题，你不必独自处理它，有很多专家可以帮助你解决这些问题。 运动也有助于充实你的精力，尽管这方面的研究尚无定论。尽量鼓励团队成员像你一样做这些事情。 首先，确保他们稳步的工作，不要有太多的加班。 然后，如果你有选择的话，请尝试在工作期间提供补充活力的健康食品，点心和饮料。

示例：保持工作和生活的平衡

我们中的许多人都喜欢自己的工作，但是，工作不是我们生活中的一切。 与改善工作的方式一样，你应该在个人生活中规划和实施一些创意，使你更快乐。 你越快乐，你的工作就越成功。 如果可以，请尝试鼓励你的团队成员也这样做。

示例：激励

激励是一个复杂的概念。 关于这个主题有一些有趣和有用的资源，也有一些不够科学。 但是，你应该了解它并持续使用它，因为它可以增加团队的活力以及提高项目取得更好结果的可能性。激励可以很简单，比如通过善意的微笑或简单的“谢谢”让人们知道你认可他们工作。 但是，你也需要小心，因为许多常见的激励形式，如货币奖励，可能会产生负面影响。

示例：协作和团队合作

善于协作的人可以创造更好的结果，但更重要的是，人类喜欢社交，并乐于成为团队的一员。 如果你可以消除团队合作的负面影响并创造一个愉快的环境，那么项目中的团队成员会更加快乐。但是，你应该小心，因为每个人是不同的，有些人需要比其他人更放松、专注和孤独的时间; 这通常是一种平衡行为。

示例：单一团队文化

不同的利益相关方都有创建或考虑小群体的倾向，并与其他群体产生紧张关系; 例如，关注项目业务方面的人员与正在构建产品的人员之间。 这种紧张感消耗了双方的大量精力，这也是我们应该努力建立单一团队文化的原因之一，每个人都朝着一个目标共同努力。

例子：群体智慧

当大量各具特色的人聚集在一起，并在一个方便协作的环境中工作时，有可能获得非常好的结果、想法和解决方案，这种结果甚至可能比来自同一类专家提供的结果更好。如果你有这样的选择，你可以定期使用它来请求团队成员帮助你解决项目中的难题。 除了获得良好结果的可能性之外，它还会使团队成员了解到他们的意见是得到认可的，他们在项目中发挥着重要的作用，从而使他们愿意提供更大的贡献。 P3.express中的活动E02是在项目中使用群体智慧的一个示例。

示例：首席项目推进者

如果你是项目经理，那么你所做的大多数事情都具有推进的特性（或者至少应该具有推进的特性）。 另一方面，你可能会发现，过去曾与项目经理有过不愉快经历的团队成员，那么这些经历会影响他们与你的关系：他们的一部分精力用于分析你对他的潜在威胁的行为，而不是与你建立信任关系，在这种情况下，你可以将你的头衔从项目经理更改为首席项目推进者。 毕竟，这就是你在项目中真正在做的事情。

NUP3

原则三：始终积极主动

我们有一种自然的倾向，那就是被动反应。 这可以帮助我们保持精力来处理不重要的事情，或者当我们处理我们完全无能为力的事情时，它可以给我们更好的结果。 这些情况与我们的项目不同，在项目中我们只能积极主动才能够获得更好的结果。

示例：规划

如果你想开车去一个新的地方，你要迟到了，为了“节省时间”，你立马出发，并在出现可能的问题时处理它们。积极主动的方法是在开始时花一些时间，比如，根据交通和可能的事故及堵塞情况，设置你的导航，以寻找最快的路径，然后才开始出发，这是在执行之前花费时间，以避免以后出现问题，从而最终节省时间。

与某些人对敏捷项目的看法不同，规划始终是必要的，只是关于不同类型计划的详细程度不同而已。执行前进行规划是一种积极主动的方法。

记住这句话：给我六个小时的时间砍一棵树，我会在前四个小时磨斧子。

如果这是一个预测型项目的话，你可能会花4个小时磨锋利你的斧头，因为你确定你要砍伐一棵树。在一个敏捷项目中，其实你不确定你是要砍伐一棵树、收集枯枝，收割草皮，采煤还是做其他事情。尽管如此，你仍然需要对所有这些事情进行一般性准备（比如，了解最近的五金店在哪里），并在你打算专注于某个解决方案时进行特定的准备（磨斧子）;这就是规划。

示例：对规划进行规划

对我们执行项目的方式进行规划是一种积极主动的方法。 这种主动性可以通过

对我们打算规划执行的方式进行扩展; 这是PMBOK®指南的管理计划、PRINCE2®的管理策略以及DSDM®中的方法的概念。有也有人将其称谓治理。

示例：持续规划

现实很少符合我们的计划，这没关系 - 但是，我们必须不断调整我们的计划，以确保它们仍然是切实和可行的。 我们应该在需要调整时尽快调整，而不是在我们遇到问题时才考虑。 这是一种主动的方法。

示例：风险管理

风险管理的整个概念是基于主动性的：当面对不确定的事件时，我们不是等待着，看到底发生的什么样的事情，然后才作出反应，而是要考虑其概率和影响，考虑应对措施，并在事件发生之前做些什么。 请注意，我们在项目中所做的事情是严肃的; 它有时是事关人们的生命 。

示例：定义角色和职责

你可以让项目团队成员在没有明确角色和职责的情况下工作，迟早会出现某种形式的角色和职责，但是代价非常高而且效果可能不好。积极主动的方法是尽早定义角色和职责，并根据需要进行调整。 这使得每个人的工作变得更轻松，他们可以专注于提供某些东西，而不是决定谁做什么。

角色的数量和种类取决于项目的类型和规模; 它可以是一个简单的定义，例如Scrum中的定义，比如P3.express中的定义，或者像DSDM®和PRINCE2®中那样很全面的定义。 但是，不要忘记这些方法中的角色描述仅涉及管理活动，你还需要为技术方面的活动添加角色描述。

示例：可用的选择

你是应该提前收尾项目还是继续开展项目？，很少只有两种选择，即使问题给出了暗示。 在做出决定之前，你需要采取积极主动的方法并考虑所有选项。 也许你可以重新审视这个项目; 也许你可以暂停，直到其他一些事情变得清晰; 或者你可以改变项目方法（例如，外包）等。

示例：批判性思维

我们都有许多偏见，这一方面有助于我们的生存，另一方面也会蒙蔽我们，导致做出错误的决定。 在做出关于项目的重要决策时，最好暂停一段时间，并考虑所有可能在问题出现之前影响我们的决策的偏见。 作为参考，你可以使用维基百科中给出的认知偏差列表： https://zh.wikipedia.org/wiki/認知偏誤列表

甚至可以使用决策框架来做出更好的决策。 起初，使用它们可能会令人分心甚至烦恼，但很快你就会习惯它们，并在没有太多意识努力的情况下从中获益。

示例：透明度

我们不喜欢项目延迟到或遇到任何其他问题，但这并不意味着我们应该隐藏它。 你应该让他透明，让利益相关方知道，因为他们中的一些人可能能够帮助你，而且他们迟早会知道这个问题及其后果，其中一些可能需要他们的早期行动（例如 ，接受负面后果）。

示例：有效沟通

在很多情况下，你可能会向利益相关方发送报告，但他们并未向您提供任何反馈。 你可能认为一切顺利，因为没有负面反馈，尽管可能不是这样。 你必须积极主动并检查他们是否真的使用了报告以及是否达到了目的，并使用这个输入来调整你的沟通方式; 否则，这个隐藏的问题可能会在以后成为严重问题，到那时就太难解决了。

示例：承担责任

很容易将糟糕的结果归咎于其他人。 例如，你可能希望你的组织赋予你所有权限，可以更改项目中的所有内容，然后完美地执行，但是，组织不会这么做，因此项目失败了。 这不是一种主动的方法。 积极主动的方法是承担责任，并在约束范围内尽一切努力做每件事情。 你不能指望组织完全信任你，并为你提供希望获得良好的结果一切，特别是当他们看到这么多项目都失败了的时候。 你要做的就是在设定的约束条件下做出一个小的提升，用它来获取一点信任，然后，使用更多一点的资源，并在更多的限制约束条件下，获取更大一点的提升，如此执行，直到达到最佳目标。

NUP4

原则四：细节决定成败

项目的各个方面都需要关注; 我们必须对项目有一个整体的视角关注。 只关注一个看似重要的方面（例如，时间）是不够的，因为所有方面都是相互关联的，只有它们都得到足够的关注才会有效。

示例：关于最后期限（Dealline）

假设你正在为奥运会建设一些东西。有一个非常严肃的截止日期，这使得时间管理非常重要，是这样的吗？如果质量很差，以至于需要返工，那该怎么办，这将会影响时间，因此，要同时关注时间和质量。比如，你可能在项目的原始定义中列出了一个梦幻般的花园，但是你知道时间不够了，你忽略了它，只盖了一个草房子，尽管你考虑到了时间这个因素，并为此作了准备。因此，范围管理也很重要。现在，我们将范围，时间和质量都作为我们的关注点了。

你听说过这样一个示例吧，肯尼迪总统在美国国家航空航天局遇到了一个看门人，然后问他做什么工作，他回答说："我正在帮助把一个人送上月球"？在项目中这样的截止日期很难设定吧。

如果你继续思考，你会注意到项目中的每个方面都有助于时间管理，只有你关注了所的方面，你才能够提供一个可接受的、具有一定确定性程度的截止日期。

示例：选择性失明

当人们面对各种各样的方法时，有时候他们会选择性失明，并创造出一个来自不同体系的、看似有趣的一个混合体， 这通常并不当用，因为元素无法独立生效，必须要相互兼容。 在一个体系中添加或更改任何的内容都应该从整体的角度来开展。

这就是为什么我们有时候会在不同方法中看到一些矛盾的元素; 这些元素在一个体系中运行良好，而在另一个体系中就不行了。 该元素本身并没有什么对错。

最安全的方法是为项目选择一种方法论，并进行剪裁，然后在考虑整个体系一致性的情况下，谨慎地添加新的元素。

示例：反流程方法

敏捷方法所做的最好的事情之一就是引起对人的方面的关注。 与流程和工具相比，敏捷宣言为个人和人与人的互动提供了更多价值，尽管这可能不是一个公平的比较。 几乎所有的方法都说人的方面很重要，而敏捷方法与其他方法的真正区别在于人的方面是其流程的一个植入部分，而不是一个简单的建议。 因此，它不是关于人的方面和流程之间的竞争，而是关于人的方面在系统中如何被看待的方式。

毫无疑问，有些人试图通过更复杂的流程来取代人的方面，但这只是一种误用。 甚至相反的情况也存在：人们试图用人的方面取代流程，这也不会很有用。

示例：这些都是你需要的知识领域

在考虑识知领域时，你应该注意不要错过任何的方面。但到底包含了哪些方面？如果你对基础资源进行检查的话，你将收到不同的答案;但是，它们都不是真相的全部。

PRINCE2®的主题是知识领域，但这些领域只是在这个方法论中起关键作用的方面。其他的方面并没有明确说明。

PMBOK®指南不是一种方法论，可以通过十个知识领域来更好地规划。但是，这些是基于PMBOK®指南对项目的看法，对所有领域的解释，它不是中立的（不一定是中立的观点）。例如，人的方面在PMBOK®指南中没有得到很多关注。

关于知识领域的一个很好的信息来源是ICB。但是，它也不是关于知识领域的，而是关于项目所需的能力。它与知识领域没有一对一的关系，但它对识别知识领域方面确实有很大帮助。项目通用原则（NUPP）中没有知识领域清单，主要是因为它是一个元体系，而不是一个体系，还因为知识领域的分类取决于项目

的类型及其环境;例如，常规的建筑项目可能需要与创造性研究项目有不同的视角。

NUP5

原则五：先定目标，后行动

除非有明确的目的，否则你不应该做任何事情。 想象一下有两个一切都是一样的平行的世界，除了你正在考虑做的事情：这些世界将会有什么不同？ 这种不同值得花力气去做那件事情吗。

如果你没有明确的目的，而只是因为其他人都这样做，或者每个人都说这样做很重要，你才做这件事，那么对你来说可能没有真正的收益，因此你可能无法从中得到任何东西; 或者对你来说，可能有潜在的收益，但由于你没有目的，你的做法可能无助于实现这个收益。

示例：项目组合和项目群

如果你参与了项目选择和启动，你应该确保关注的是收益和需要解决的问题，而不是那些实现这些目标的产品本身。 典型的例子是电梯制造商，他们过去常常接受有关电梯速度的投诉，因此开展了多个项目以提高电梯的速度，但没有取得圆满成功。 直到他们决定关注问题（人们的无聊或不适）而不是看似自然的解决方案（更快的电梯）。 最终他们为电梯安装了镜子，简单而低成本的解决了问题。

请记住，项目管理主要关注的是把事情做正确，而项目组合管理关注的是做正确的事情。 你如何把一个项目开展好，可能无关紧要; 但是如果你做错了项目，将会一无所获。 这就是说做任何事情需要有一个目的。

示例：将项目作为一个整体看待

产品的灵活性因项目而异。在一些IT开发项目中，产品是完全灵活的，其最终形式取决于项目期间产品增量所产生的反馈，这需要采用自适应（敏捷）方法。实际上，这是项目组合，项目群和项目层级的组合，需要最大限度地关注

最终目标。将目的记录下来，并使其被方便的访问是一个好主意;这是一些Scrum项目中使用的“产品愿景”的目的之一。敏捷项目中对商业价值的关注是其确保与目标保持一致的方式。

在其他类型的项目中，产品相对固定，并且还有其他机制来确保所确定的产品来满足目的，项目团队成员可以将其大部分重点从目的转移到产品上（因此，有了PRINCE2®的“关注产品”原则），但至少需要对目的有最小的关注度，以确保正在建设的内容将能够满足目的，这可以通过将预测的收益与预期的收益进行比较来完成（因此，有了“持续的商业论证”原则和PRINCE2®中的“商业论证”主题。）当为外部客户开展项目时，对于此项目，客户有自己的目的，而你的公司会有不同的目的。你应该理解并使用这两者才能真正成功。

示例：项目监控

关注最终目的是必要的，但对于很多日常的用途来说，可能过于抽象。这就是为什么需要一个结构化的驱动方式来使其更实用。首先，项目的理由和收益是根据最终目的来定义的，然后使用明确的和隐含的项目变量目标（例如，时间，成本和质量）来满足这些理由，接下来是满足最终目的。这些是较低级别的目的，对我们的许多日常任务都很有用。

在监控方面，项目的低级别的监控将使用最低级别的变量完成，因为可能无法跟踪最终目的。在这种情况下，你仍应牢记最终的目的：监控的目的是什么？

一个正常和可接受的答案是看项目是否按计划执行，如果没有，则触发某种应对措施，这将使项目重回正轨或调整目标，看看它是否仍能满足最终目的。我们的衡量因此应该能够帮助实现这种低水平的目的，并且正确的测量结果可以对这些变量在项目完成时的值进行预测;例如，我们何时才能完成该项目？我们需要多少钱来完成这个项目？

所有其他衡量（例如迄今为止的计划值和实际值）只是计算预测所需的临时值，而不是用于管理目的的最终值。

示例：文档

无论你在项目中使用何种开发方法，计划始终都是必要的。重要的问题是每种计划的详细程度。如果计划中没有足够的细节，计划将无法发挥其作用，项目的执行将基于临时决策而非整合的整体决策。另一方面，许多人为了保持谨

慎，在他们的计划中添加太多细节，这使得计划的准备和维护变得非常困难，并且对于项目的不确定性而言过于僵化。结果，该计划变得不切实际且毫无用处。

决定计划细节水平的最佳方法是为计划中的每个元素设计一个目的。例如，如果你正在考虑为计划添加资源，那么你应该有一个目的：你打算如何使用资源？资源对该项目有什么帮助？需要多少工作量？是否值得投入？

有时候是要决定是否要在计划中使用某个元素，有时还要考虑计划或准备某些内容的方式。无论是商业论证，项目章程还是报告：你应该问自己为什么要准备这个文档以及对项目有什么帮助。

基于“模板”与基于目的做某事是完全相反的。

示例：状态报告

在许多项目中，通常会有一个非常长的状态报告。 基于项目通用原则（NUP），我们应该问自己报告的目的是什么，以及我们如何能够实现这一目的，而不是关注其他人可能在做什么。

在许多情况下，这使得我们会准备了一份非常简单的、利益相关方会真正阅读和理解的单页报告，而不是一份长报告。 这是对目的的关注。

但是，如果你只准备一页长度的报告，有些人可能会就此不满，并认为你没有的提供“适当的”监控系统。 在实践中，这为你创建了第二个目的（除了第一个目的，帮助利益相关方了解项目的状态），为了满足这一目的，你可以简单地创建非常长的第二类报告。 但是，你不会将其与第一个报告混合，因为这两个目的不同。

示例：商业论证和项目章程

准备诸如商业论证和项目章程之类的文件对于大多数人来说通常是无聊的和盲目的，对于大多数人来说又是官僚主义的必需品。而这些文件又适用于大多数项目。 如果你试图找到一个“模板”并进行填充，那么工作就浪费掉了;你可以检查这些文档的真实目的，看看它们是否以及应该如何在您的项目中发挥作用，然后以你喜欢的任何方式来准备文档，只是为了满足“这将会是正确的文档”这个目的。

当你在考虑如何准备这些文档以满足其目的的时候，你可能会想不到每一个场景，可能会遗漏某些内容。 为避免这种情况，您可以查看PRINCE2®，PMBOK®Guide，P3.express和DSDM®等建议的资源，然后根据目的评估这些建议。

示例：项目后

虽然项目有特定的目的，而且，我们可以在整个项目中考虑这个目的，但目的的实现主要是根据项目期间的预测来评估的。当项目完成后，我们不应该忘记这个目的。在项目完成后检查目标的实现很重要，因为我们可以看到最终目标是如何实现的，并为未来的项目提供经验教训，有时，在项目完成后，只有我们执行一些额外任务时才能实现目标，并且要理解这些额外任务需要监控的必要性。项目后监控是目标驱动的必要步骤。这主要是项目群和项目组合管理体系的责任，由于大多数公司没有这样的管理层，这一步骤通常被忽略。这就是P3.express和DSDM®等方法将这一步骤添加到项目管理方法中的原因。

示例：简单

世界是复杂而混乱的，但我们的模型是世界某些部分抽象近似的表达，因此它们可以很简单。 另一方面，简单系统通常效果更好，因为我们可以专注于主要想法。

我们中的许多人试图通过增加系统的复杂性来获得更好的结果，希望它能够匹配世界的复杂性，在实践中这使得系统难以使用，并且通常阻碍了我们目的的实现。

示例：剪裁

用于流式传输的音乐的软件与用于医院或飞机设备上的与人的生命休戚相关的软件完全不同，与用于卫星当中，计划使用多年且不会发生故障的软件也不同，也不同于建造一个别墅，一个消防站和一个加工厂。当你考虑到这些不同的目的时，你会更好地了解如何为不同项目剪裁体系和工件。

NUP6

原则六：使用可重复的元素

对于一个项目来说，临时的方法需要消耗太多的精力和资源，并且总是存在遗漏一些必要元素的风险。 简化必须完成的工作的最佳方法是使用可重复的元素，并且最好可以重复循环使用。

示例：质量检查单

一个检查单是许多人在个人和职业生涯中使用的，一个潜在可重复元素的简单示例。以可交付成果的质量标准为例，例如：

- 首先，你可以创建一个所有标准的清单，这是一种计划形式。
- 项目通用原则六（NUP6）的建议是尝试对其进行归纳：项目中是否还有其他类似的可交付成果？在这种情况下，为该类可交付成果准备一份通用质量检查单，并将其用于所有质量检查单。如果有不同的地方，请保留通用清单，并为各个单独的可交付成果添加一些额外检查项。这样你就有了可重复的质量检查单。
- 一旦为各种类型的可交付成果准备了通用检查单，你可能会发现在它们之间重复的元素，这表明需要为它们准备一个虚拟的上级父类别。在这种情况下，你可以提取它们并将它们放在上一级的检查表中，而不是在通用检查单中重复所有这些的项目。最后，你可能会为整个项目提供一个通用的检查清单。 Scrum中对“完成”的定义是使用项目级检查单来提高质量的一个例子（可能包括其他内容）。通过这样做，每个可交付成果将属于一个类别层次，应该能够满足所有类别清单中出现的项目。

据此，上一级检查表中的项目对其下的所有可交付成果是可以重复使用的，这样可以节省计划和执行的时间与精力。

更重要的是，一旦你为一个项目执行了此操作，你就可以进行剪裁，将其应用

于将来的所有类似项目中，这是一个对多个项目进行规划的可重复的形式。

示例：流程和工作流

一些可交付成果或与之相关的目标需要一些特定的步骤，这些步骤是标准化和可重复的。例如，如果交付物需要进行独立设计和批准，你可以准备一个简单的工作流，使所有步骤，涉及的人员和大致的持续时间都很清晰，从而避免许多困难。但是，你应该小心，不要让流程过于复杂或文档过于密集繁多，因为这会适得其返。参与项目的所有人都应该能够看到，工作流和流程可以为他们的工作提供支持，使他们工作起来更容易，他们需要的不是阻碍他们实际工作的官僚化的文档。

敏捷项目在其迭代开发方法中具有可重复的元素，其中每个特性都重复某些类型的开发活动;例如XP（极限编程）中常见的惯例是：配对，挑选项目，在白板上设计，构建测试脚本和代码，代码集成等。

除了可用于技术活动的可重复工作流之外，你还可以为项目管理活动提供可重复的元素。 PMBOK®指南，PRINCE2®和DSDM®中的流程，P3.express中的活动以及Scrum中的事件都是这个概念的示例。

示例：周期

拥有可重复元素对于管理项目来说是有帮助的。 通过将它们置于可重复的周期中，可以使管理项目变得容易。 这些周期大大简化了参与项目管理和领导的人员的日常活动。 用于具有多个阶段项目的PMBOK®指南中的过程组周期，PRINCE2®中的阶段，P3.express中的每日，每周和每月周期，DSDM®中的迭代和时间框以及Scrum中的Sprint都是这个概念示例。短的周期比长的周期更容易理解和使用; 例如，与PMBOK®指南的阶段相比，Scrum中的Sprint更容易理解和使用。 但是，太短的周期可能不足以支持某些类型的项目，解决方案可能是使用多个周期，例如DSDM®将较短时间盒周期与较长的迭代周期一起使用，或者P3.express中使用每日，每周和每月周期。

示例：方法

使用一个方法论或一个框架来开展项目是可重复元素的另一种使用。 这可以是

现存的体系，如PRINCE2®，P3.express，DSDM®或Scrum，也可以是你自己定制或构建的体系。 但是，从一个现有方法开始，并根据你的需要进行调整，而不是从头开始构建，通常是一个更好的主意。

任何可重复的元素都是抽象的，需要定制才能使其适应现实世界。 然而，有一系列的抽象和定制需求：小的，相对具体的质量检查单处于这个系列的一端，具有少量的抽象和剪裁需求；而方法论的另一端，具有较高的定制需求。 你应该始终注意剪裁的需求，否则，可重复元素将无法正确的满足你的需求。