

无论具体活动或项目网络本身如何，这些软件工程最佳实践将加速整个项目。它们在任何项目、任何环境和任何技术中都是有效的。虽然用这种方法改进项目可能显得会提升成本，但最终很可能会降低成本。开发系统所需的时间的减少支付了改进的成本。

9.2 进度压缩

之前的进度加速技术列表中的条目存在的问题是，没有一项可以真正快速起效；它们都需要时间才能发挥作用。但是，可以做两件事来立即加快进度，要么使用更好的资源，要么找到并行工作的方法。通过使用这些技术，可以压缩项目进度。这样的进度压缩并不意味着更快地完成相同的工作。进度压缩意味着更快地达成相同的目标，通常是通过做更多的工作来更快地完成任务或项目。我们可以将这两种压缩技术结合使用或单独使用，对项目的各个部分、整个项目或单个活动都可以使用。这两种压缩技术最终都会增加项目的直接成本（稍后定义），同时减少时间。

9.2.1 利用更好的资源

高级开发人员将比初级开发人员更快地交付他们的工作部分。然而，有一种常见的误解，认为这种差异是因为他们的编码速度更快。通常，初级开发人员的编码速度比高级开发人员快得多。高级开发人员花在编码上的时间越少越好，相反，他们花在设计代码模块、交互和测试方法上的时间越多越好。高级开发人员就他们正在使用的组件以及所使用的服务编写测试装置、模拟器和仿真器。他们记录自己的工作，考虑每个编码决策的含义，并研究服务的可维护性和可扩展性，以及其他方面，例如安全性。虽然在单位时间内高级开发人员比初级开发人员编写的代码少，但是他们完成任务的速度更快。就像我们所知道的，高级开发人员比初级开发人员的薪资要求更高。应该将这些更好的资源分配给关键活动，因为分配到非关键路径不会改变进度。

9.2.2 并行工作

通常，每当进行一系列活动并找到同时执行这些活动的方法时，都可以加快进度。这里有两种并行工作方式。首先是通过拆取活动的内部阶段并将其移到项目中的其他位置；第二种方法是消除活动之间的依赖关系，以便可以并行处理这些活动（如第7章所述，将多个人同时分配到同一活动是行不通的）。

1. 拆分活动

我们可以拆分活动，而不是按顺序执行活动的内部阶段。可以在活动之前或之后，将一