

①辅助或代替分析阶段：在分析阶段利用快速原型方法可以得到良好的需求规格说明。在整体上仍然采用传统的模式，但使用原型化方法来补充和完善需求说明以达到一致、准确、完整、无多义性地反映用户要求，从而代替了传统的仅由复审和确认来提高需求规格说明质量的方法。并能在早期克服潜在的误解、遗漏和错误，尽量不让潜在的问题遗留到开发的后期，减少将来维护的代价。

②辅助设计阶段：在设计阶段引入原型，可根据需求分析得到的规格说明进行快速分析，得到实现方案后立即构造原型，通过运行，考察设计方案的可行性与合理性。在这个阶段引入原型，可以迅速得到完善的设计规格说明。原型可能成为设计的总体框架，也可能成为最终设计的一部分或补充的设计文档。

③代替分析与设计阶段：这时不再遵循传统的严格按阶段进行软件开发的要求，而是把原型方法直接应用到软件开发的整体过程。在实施原型开发的过程中，不再考虑完善的需求说明，把分析、定义和设计交织在一起，通过原型的构造、评价与改进的迭代过程，逐步向最终系统的全面要求靠近。由于在分析的同时也考虑了设计与实现的要求，能更有效地确定系统的需求和设计规格说明。

④代替分析、设计和实现阶段：在软件开发环境的支持下，通过原型生存期的反复迭代，直接得到软件的程序，交付系统测试。这属于进化型的原型开发，由初始的基本需求得到最初的原型开始，一直进化到软件的整体系统，并满足用户的一切可能的要求。

⑤代替全部开发阶段：这是典型的进化型原型开发方法。完全摆脱了传统的软件生存期模式，通过反复的原型迭代过程，直接得到最终的软件产品。系统测试作为原型评价工作的一部分，融入原型的开发过程。