# 原型风险 （书p275）

虽然原型可以降低软件项目失败的风险，但原型本身也有风险。

原型风险可归纳为以下四点：

1. 原型发布的压力
2. 受细节所累
3. 不现实的性能预期
4. 对原型投入过多

**原型发布的压力 （书p275）**

原型发布的压力很多情况下存在于可抛弃原型中。可抛弃原型仅仅是一个模型，一次模拟，一个实验，它不能用作产品。而很多原型评估人员或客户会误当作是实际产品。

发布这样的原型很可能会导致项目延期完成，因为原型的设计和编码并没有考虑到软件的质量和生命周期。

解决方法：

1. 应该要让每个关注原型的人都要理解原型的目的及局限性，清楚创建原型的原因，并且明白我们才是决定原型最终命运的人。
2. 提交纸质版的原型，而不是电子版。
3. 选择合适的原型开发工具，使得原型看起来简陋些而非完善。

总之，不要畏惧于提交不成熟的产品的压力而创建原型，让看到原型的所有人都明白它将来不会作为产品软件发布。

**受细节所累 （书p276）**

原型的另外一个风险是用户把注意力放在与UI有关的外观和操作细节上。如果使用一个看似真实的原型，用户很容易忘记自己还在需求阶段，应该关注与概念相关的问题。

解决方法：

将原型限定于显示画面、功能和导航选项，可以消除不确定的需求。

**不现实的性能预期 （书p277）**

第三个风险是用户根据原型的性能来推断最终产品的预期性能。不要在预期产品环境中对原型进行评估。模型使用的工具或语言和实际产品开发的环境在效率上会有差异。

比如，概念证明型原型可能没有安全层，在查询模拟数据库时使用硬编码的查询结果示例，查询响应就很迅速，从而导致评估人员以为产品在访问大型分布式数据库时也有同样的优异性能。

解决方法：

在构建原型时要考虑时间上的延迟——或者让原型看起来还没有准备好马上发布。并在屏幕上显示信息，申明它并不代表最终产品。

**对原型投入过多 （书p277）**

不要在原型工作上投入太多精力，否则可能会导致开发团队没有时间而不得不将原型作为产品或者匆忙进入混乱的产品实现。

典型的情况就是：对整个解决方案进行建模而不是只对最不确定的、高风险或复杂的部分进行建模。

如果原型能够测试假设，能够回答问题并且能够提炼需求，就足够了。