### 未来软件抗衰的关键问题与发展趋势

从上一节可以看出，软件抗衰的第一层含义一定程度上已达到，第二层含义尚未实现，因为现有的抗衰方法一般基于特定的软件衰退问题（如内存泄露），而软件衰退的偶发性和不确定性使这些方法常常有漏洞。沿着软件抗衰方法发展的轨迹，本文认为未来的软件抗衰方法，应该解决如下几个关键问题：

1. **动态评估软件健康状态，自适应确定预测关键点**

将软件不健康的征兆作为开始预测软件剩余寿命的条件，应该考虑动态多变的环境的影响，动态评估软件的健康状态，区分正常的负载改变、软件升级等行为引起的性能变化和真正的性能衰退，解决基于模型和度量的混合抗衰方法中，预测关键点固定、不适合动态多样的软件衰退的问题。

这涉及到分析衰退对软件系统的影响，刻画运行中软件的健康状态、量化和评估衰退程度等问题，这些问题是动态确定预测关键点、制定和实施抗衰策略的重要依据。目前的软件抗衰方法主要通过检测系统资源的耗费情况、响应时间和响应率等来验证软件衰退的发生，但不能进一步量化。

1. **自适应预测软件整体剩余寿命**

软件剩余寿命即为当前时刻到软件不可用（或失效）的时长，未来的软件抗衰方法在预测软件剩余寿命时，应能应对未知的软件衰退模式，包括偶发性强的、衰退因素未知的、模式动态、多变的软件衰退，它们的共同特点是衰退样本缺乏或难以获取。现有抗衰方法对剩余寿命的预测依赖于衰退样本，且一般基于特定的软件衰退情况（如内存泄露）进行预测，选取的模型片面地拟合已有衰退样本。这些方法在预测偶发性强、动态、多样的软件衰退导致的失效时，易发生因缺乏衰退样本而被逃避的问题，因此需要构建不依赖衰退样本即可对未知衰退模式影响下的软件的整体的剩余寿命进行预测的方法。

1. **群体抗衰机制**

当前的应用软件越来越多地以集群方式工作，因此，在（网络）应用软件系统中，每个节点的应用软件的抗衰系统就不能各自为阵、不顾其他节点上软件的健康状态，而应该通力合作，以把握系统中软件衰退的整体趋势，并据此相互协助、取长补短，以提高整个软件系统抗衰的效率和软件系统的整体可用性和性能。