



# 生命周期自动化流程：从概念到实现

**Louis J. Taborda**

**Chief Technologist,  
Lifecycle Management**

## 目录

简介.....	3
<b>使流程自动化的动机 .....</b>	<b>3</b>
生命周期的复杂性.....	4
生命周期中的孤岛 .....	4
<b>需求管理自动化.....</b>	<b>5</b>
使开发阶段顺利衔接.....	5
减少管理的工作量.....	5
主动流程的基础结构.....	6
<b>在现实中进行测试.....</b>	<b>8</b>
开发过程的共同特征.....	8
从技术角度来考虑.....	9
<b>生命周期自动化工具：现状报告.....</b>	<b>11</b>
<b>参考文献.....</b>	<b>11</b>

## 简介

对开发生命周期的典型说法是把它看作系统开发的不同阶段的总和，但这样做会使开发者很难评估生命周期对实际工作的真正影响。

即使许多开发组织正致力于使生命周期的各个阶段集成与自动化变得更好，但他们还不清楚怎样达到这些目标。他们不得不保留分阶段式的设计方式，陷于他们已有基础环境的限制并且无法减缓典型开发项目的压力。

传统的开发工具可以提供非常有价值的支持，但是他们太局限于单一的阶段，只能为这一阶段的工作带来利益。使各阶段相互协调与顺利衔接的需求必须得到重视并形成系统化、程序化。

使整个开发生命周期自动化的目标是非常吸引人的，生命周期自动化能使发布周期更快、更可靠。消除不同阶段工作的隔膜与减缓支持专业领域的负担可以使组织更好地管理日益复杂的开发项目并实施最佳经验，如果没有生命周期自动化工具，最佳经验就变得难以实施。

只有实际地解决了开发人员每天为满足客户日益增长的需要所面临的问题，生命周期自动化工具才能称得上成功。为了更好地理解实现生命周期自动化所面临的挑战，本文将阐释当今开发的几个共同特征与导致开发项目陷于混乱的根源。

值得庆幸的是，随着象 Telelogic 这样的公司所提供的生命周期工具的日益成熟，这些问题有望能得到解决。本文介绍的这一技术将最终实现对系统开发项目的生命周期的支持。通过讨论新工具可以显示出生命周期自动化工具正在逐渐成熟，新的工具将帮助解决前面提到的不同情况的开发问题。

## 使流程自动化的动机

因为其它行业已经实现了流程自动化，这些技术方案使得那些行业得到了快速发展与繁荣，所有不难理解为什么会有这么强烈的动机来实现开发流程的自动化。

在开发早期，开发人员并不缺少工具，但是这些只是单一的工具，只支持特定的工作或任务。当人们力图把它们扩展到整个开发生命周期中去的时候，就超出了这些工具的范围。早期的计算机辅助软件工程 (CASE) 工具就属于这一类工具。

当今，工具行业正处于一种有利的位置来处理整个开发流程的问题。正如 Telelogic 公司自己的历史，随着单一工具供应商的相互合并将为整个开发流程自动化提供基础。当然，前途依然充满风险，但在当今开发所面临的艰难条件下，做这种努力是绝对必要的。

## 生命周期的复杂性

随着客户对系统的要求越来越复杂，建立这些系统的流程也随着变得更复杂。虽然现在出现了许多不同的开发流程与方法，但是广义地说，它们分为两个阵营：传统的开发流程与敏捷开发法 [1]。令人感兴趣的是，敏捷开发法是以业界没有能够很好解决的部分流程做为例子来说明非常细化与复杂的传统流程。

来源于软件工程学院 [2] 的能力成熟度模型提供了对许多种生命周期流程的全面描述，可以作为建立成熟的开发组织的参考。虽然有许多公司采用 CMM 来改进他们的流程并取得了良好效益，但对很多开发团队来说，实施 CMM 是相当困难的。有些人说 CMM 所要求的复杂流程对小组来说是不现实的——这样的流程要求太高，很难被用于实践。对生命周期的必要流程进行自动化往往需要改变现有的工作方式。

## 生命周期中的孤岛

当今的项目开发需要不同团队的专家进行合作，在大型项目的开发中这种合作本身就是个大问题。即使在认识到开发的瀑布式模型之前，开发各个阶段的专业划分导致流程与技术过于细化使得开发周期的各阶段成为孤岛。

专业化导致在不同的生命阶段使用单一的技术、语言、概念、构件、格式与工具；它们并不都能够被容易地衔接来支持生命周期。开发的各阶段通常是在专业的壁垒中进行的，由各阶段的专业工具来支撑，从而导致信息的孤岛 [4]。包括系统工程、需求管理与配置管理在内的生命周期管理技术与领域，只有在后期才被引入来形成全局与系统的概览。

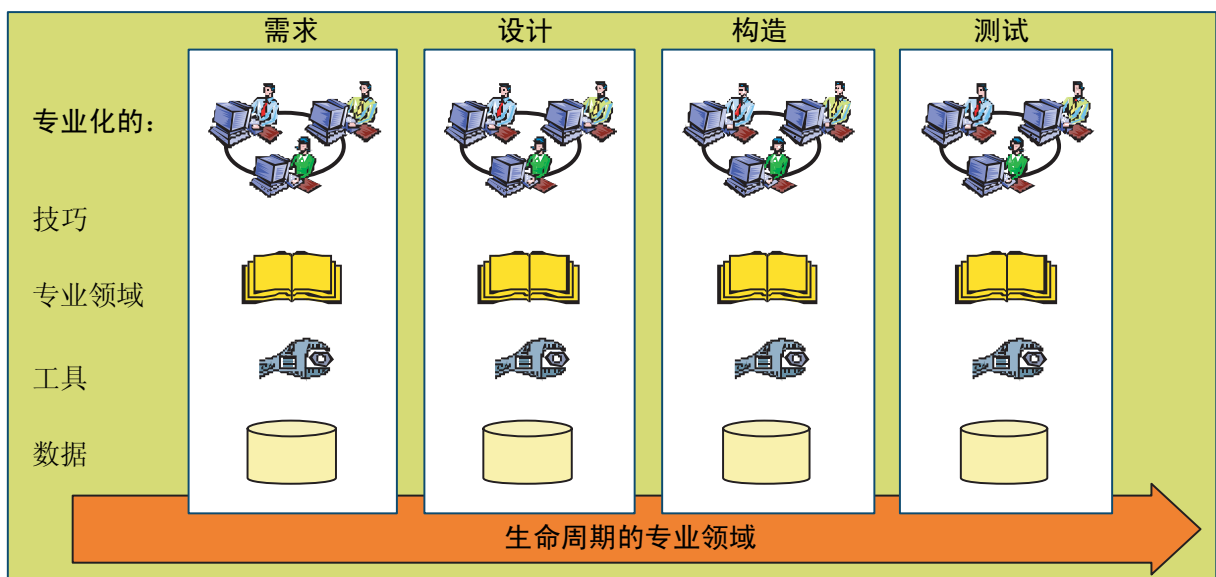


图1 基于阶段式的优化所造成的开发竖井，必须使用专业的生命周期管理才能维护整个项目。

当今的大项目通常由几个工作在特定阶段的专家组来开发。每个小组都有独特的、包含自己语言与基础的文化。这些团队不得不由生命周期支持专家组来管理，由这个专家组作为裁判来决定是否每个小组的工作都达到了项目预期结果。每个专家小组都力图在自己独自的范围内对流程进行优化以获取现阶段的成果。生命周期支持专家组与在各阶段工作的小组之间经常产生矛盾，这并不令人感到奇怪。应该使用生命周期自动化工具来打破这些隔膜从而恢复不同小组在整个生命周期内的和谐。

## 需求管理自动化

对于任何系统开发，在自动化之前，你必须充分了解需求。对这一领域必须有深入的了解并弄清楚对解决方案的期待。生命周期自动化这一词的真正含义是什么？隐藏背后的商业动机又是什么？

使整个生命周期实现专业化与对开发流程的需求是驱动生命周期自动化的关键因素。自动化的议程有可能改变现有的开发方式，而不仅仅是使那些现有的不完善的流程自动化。概括地说，整个开发流程的自动化可以获得以下几点益处。

### 使开发阶段顺利衔接

业界有足够的动力来架起不同开发阶段间的桥梁，已使生命周期各阶段的衔接更加紧密。当 CMM 与其它工业标准正在定义生命周期流程的时候，对象管理组织（Object Management Group）的模型驱动构架与 UML 2 标准正把可视化开发推向新的层次来支持更多的生命周期过程。

这代表了另一种最终只需“按一下按钮就可以生成系统”的远景。虽然从需求直接生成整个系统还可望不可及，但是可视化开发工具已经取得了长足的进步。Telelogic Tau®提供了一套工具能够从可视化模型生成关键的源代码，从而减少了贯穿生命周期不同小组之间的沟通障碍，使从一个开发阶段过度到另一个阶段变的更容易。

使不同阶段的小组之间顺利衔接可以缩短上市时间，并且自动化能成为保证必要的流程步骤被执行的一种方法。相关的信息能够被传递到下一环节的专家小组中；转换成适合他们需要的信息。同一信息可以被自动记录与管理以便日后使用，相关状态也能够马上被更新。

从本质上讲，生命周期自动化工具可以打破开发各阶段的隔膜（正如企业管理系统帮助打破业务上的隔膜一样），实现在生命周期各阶段间的合作与信息共享。

### 减少管理的工作量

改善各阶段间的过渡可以减少隔膜与大大地改善效率，同时生命周期自动化也为应用

生命周期的专业知识与最佳实践提供了很大的便利。

支持这些不同阶段的专业领域是生命周期自动化工具与传统单一工具的本质区别。虽然一些基本的工具集成有助于各阶段的衔接，但要支持整个生命周期的流程则需要更复杂的技术。

这种困难源于这些领域的投资回报并非是立竿见影的，实施人员的努力也很难得到回报。这确实需要专业的技能，但不幸的是，这些过程仅仅作为各种值得考虑的最佳经验，尤其是当压力很大的时候。典型的例子是在整个生命周期中对需求进行跟踪。充满压力的实施人员不可避免地会认为这是他们实际工作中的琐事。这些是可以理解的，因为他们的努力并不总是能在生命周期的后期被明显地表现出来。

通过使开发流程自动化，这些生命周期管理就无需花费什么精力，使最佳的经验实施起来更容易，并能在组织内得到更广泛的推广。

原来需要开发人员或支持工程师大量管理的工作，现在通过生命周期自动管理工具就可以简单地完成。实施人员再也不必被那些需要使用不同的工具来完成地琐事所打扰，他们可以集中精力做他们擅长的事，必要的管理工作可以由自动化的工具来完成。

Telelogic *ActiveCM*<sup>™</sup> [7]，适于软件开发人员的自动且透明的配置管理（CM）解决方案就是这一技术的很好例子。它可以使用户摆脱繁琐的流程。在这里，需要小心从事的存入/取出操作由 CM 工具自动在后台完成。

生命周期自动化工具可以减少实施象 CMM 那样最佳实践所需的管理负担。今天的用户不再需要那种对同样的数据还得进行重复输入的系统。生命周期自动化可以使开发组织改进他们的流程而无需繁重的管理工作。

### 主动流程的基础结构

已经有许多标准、书与开发过程可以提供怎样提高开发效率的指导，没有必要再给出更多的建议了。大多数有经验的开发者只知道“应该做些什么”；但在实际的开发项目中并没有机会去做这些事情。由商业驱动的紧迫的开发项目很少提供这种机会与资源来实施哪怕是一小部分“应该做的事情”。

在书中或网页上所描述的最佳实践或流程仅仅是一种被动的知识。这种指导必须被应用，当然这才是真正具有挑战的地方。CMM 已经免费推广了十多年了。特别是对小的开发小组来说实施生命周期流程还很难，要到达被动流程所描述的能力成熟度还有很长的路要走。

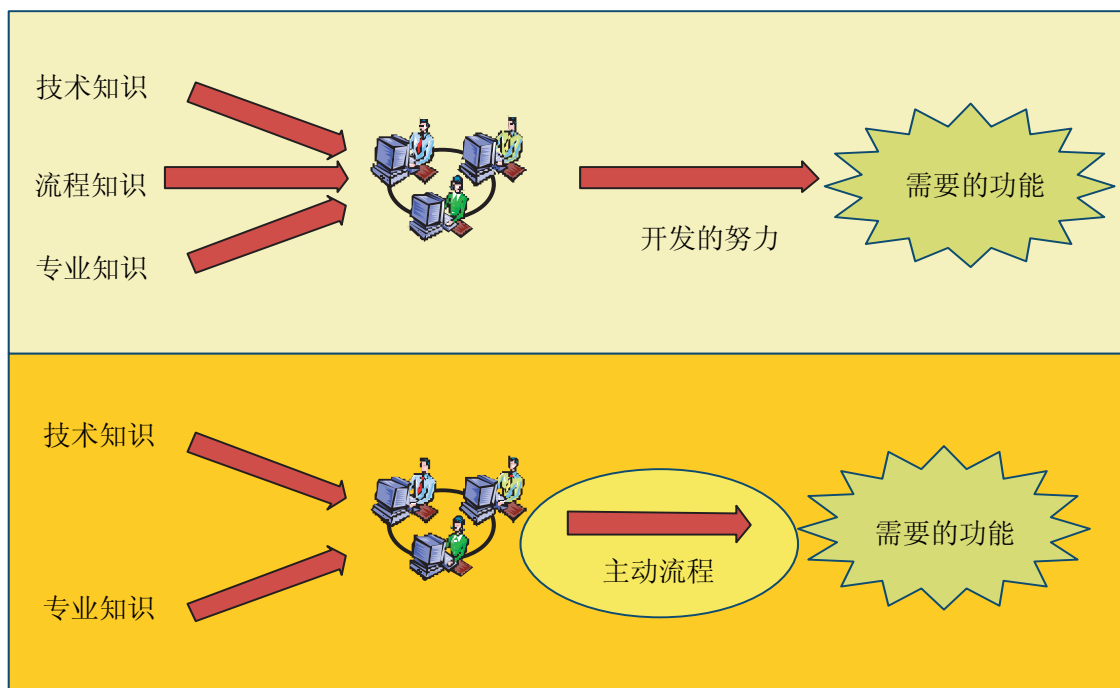


图2 生命周期自动化应该支持一个主动的流程，可以把精力集中于开发本身。内嵌的流程知识可以容易且快速的生成所需系统的功能。

生命周期自动化工具可以帮助实施这些最佳经验并为主动流程提供实施的基础。主动流程是通过在应用中支持实施人员，把被动的知识应用到实际的工作流程中与并使信息共享的流程。

对主动流程支持工具可以实现其它方法难以实现的流程。能够支持这种改进的流程的自动化生命周期工具具有如下特点：

- 主动生命周期的基础构架能够容纳不同的流程，对不同大小、不同条件的开发项目可以灵活使用；
- 维护生命周期的内容包括自动且透明地支持生命周期的各个领域使它们更容易实施；
- 对流程的支持已经被嵌入到支持生命周期自动化的工具中，使它容易执行，即使在紧迫的时候也不容易被忽略、逃避或遗忘；
- 对偏离所选择的流程很敏感，可以辨识容易忽略的步骤并为日后修补作出标识。

## 在现实中进行测试

### 开发过程的共同特征

对自动化的生命周期概念的最终测试是看它在实际环境中的能力及给出的解决方案是否能解决实际问题。

下面是代表当今实际业务环境中项目开发的七个共同特征。经验表明以下的条件是导致许多项目开发混乱的祸根：

- **不协调的团队**

前面已经提到，项目开发需要把逐渐增加的参与者组织到不同的队伍中。对不同的队伍实施生命周期管理是一项很具有挑战的工作。

- **分阶段的发布**

现在很难找到一个项目仅做一次系统发布。今天的系统复杂性与流行的迭代式开发结合在一起，甚至融合了基于部件的开发技术，这意味着在任何时间内都没有一个单一的生命周期能够贯穿始终。每一次发布，无论是对部件还是整个系统或是补丁，都有它自己的生命周期，使得沟通与对项目的评估变得复杂。

- **值得质疑的状态**

对开发活动状态的评估非常不可靠。一般来说，需求与最终给出的结果之间的关系总是很模糊，当然也很难确定状态。今天的项目状态报告也仅仅是项目负责人主观臆断的结果，与实际的工作情况关系不大。

- **无处不在的变更**

在生命周期中变更随时都可能发生，并非通常想象的一劳永逸。变更的不确定性与其所产生的一系列不确定的影响是项目混乱产生的根源之一。

- **模糊的范围**

无论是在传统的国防部门获取项目或进行动态的、基于部件的网络开发，项目的复杂性都使得系统需要被分解。如果不能定义清楚系统部件的界限与接口，开发就会演变成一种混战，导致各方都精疲力竭。

- **阶段惯性**

前面已经讨论过，在新阶段开始时进行必要的阶段衔接可以减少交接时间。但是在新团队能够承担任务或熟悉工具之前的任务转换与熟悉所需要的时间是必要的。随着迭代时间缩短的趋势，总的项目交接过程会大大消耗生产率。

- **稳态集成**

系统被分解之后还需要重新集成。当把瀑布式开发模型应用于复杂的系统开发，



会导致后期集成工作的大量增加，而迭代式开发与前面提到的分阶段发布会可以使开发团队处于连续的、稳态周期的集成与测试之中。减少项目风险可以带来利益，特别是在每个周期中都发现新问题与错误时，项目波浪式前进会需要各种资源与管理。

### 从技术角度看问题

今天，工具使得许多关键的开发项目变得可行。对开发新的尝试，如非常编程（eXtreme Programming）[6]，能处理大规模协作与更广泛的流程模型，可以改进贯穿生命周期的协调。开发问题的绝对尺度（开发者每日工作的混乱度）要求组织接受与采纳这种技术支持与自动化过程。

虽然以上情况看起来有些不尽人意，但新的生命周期技术正在解决这些问题。通过降低过程改进的难度与提高能力成熟度，生命周期解决方案支持象 CMM 这样已经确立的但需要大量手工的流程框架。

下表显示了 Telelogic 在一段时间内是怎样面对这些问题的。现有的解决方法将会扩展到支持整个生命周期自动化过程。

特征	所需的技术	Telelogic 自动流程的支持
不协调的团队	主动流程的基础可以帮助实施一个灵活的、贯穿生命周期的一般性流程。	通过DOORS & Synergy来实施的需求驱动开发方式支持团队使需求与产品保持一致性。
	共享的数据仓库可以管理与控制从开发初期到发布的所有生命周期。	Synergy的数据仓库可以管理整个生命周期的代码并与DOORS集成来保证需求与代码相关。
被质疑的状态	每一次的发布都被分解为变更任务以便能够直接衡量实际的进展情况。	Synergy的基于任务的变更管理支持从发布状态到委托实施任务的管理。
	数据仓库技术提供的可跟踪性能够管理从最初需求、测试与发布的整个生命周期的进展状态。	DOORS 提供需求的覆盖与最终发布产品的关联关系的管理，Synergy的状态报告可以显示每次发布的状态。DocExpress 提供自动文档生成，为集成的生命周期提供了完整的视图。
分阶段的发布	支持多次、并行的发布，为不同的生命周期阶段提供监控。	CM Synergy 的流程模型支持发布规划与管理，可以对底层的开发任务进行管理。
	提供对单个部件发布的跟踪及其与整个系统发布之间关系的框架。	CM Synergy通过对系统分层结构的支持，可以支持单独的部件发布。
无处不在的变更	一个通用的可裁剪的变更管理系统提供对贯穿项目始终的用户、业务、修改的变更一致性处理。	ChangeSynergy提供企业级的变更管理可以满足各种变更要求。
	灵活的变更工作流程可以使不同的流程应用于不同的项目与不同的目标—改变需求要冒很大的风险。	DOORS 企业级变更建议系统为控制变更请求所引起的需求改变提供了灵活的机制。
模糊的范围	系统分解支持对每个系统子模块与接口的分层次的详细描述。。	DOORS提供对分层结构描述的全面支持并可维护它们之间复杂的关系
	对需求的描述可以用于驱动随后的开发任务并保证开发在所定义的范围之内。	需求驱动式开发提供了保证被分解的任务是围绕需求进行的手段。
阶段惯性	可视化工程与模型驱动构架提供了从分析到设计及实际实施与测试的无缝过渡。	在Tau 中建立的UML/SDL模型能够自动生成可执行的代码。使用与DOORS的接口可以保证模型是根据需求开发的，同时测试用例也是从高层的描述生成的。
	自动化流程支持从一个开发阶段的配置到下一个开发阶段的配置转移。	CM Synergy的灵活的基于任务的流程自动化支持软件系统部件从开发到发布的整个过程。
	部件装配与集成的基础框架可以使所选取的部件迅速生成系统。	CM Synergy 通过支持项目嵌套与子项目提供逐步集成的方法。
稳态集成	自动的环境设置可以对配置更新与快速变化进行集成。	CM Synergy 的编译管理能力允许对工作空间/环境的快速刷新，允许快速编译/测试。

表1 Telelogic现有的解决方案已经在一定程度上提供了对生命周期自动化的支持。

## 生命周期自动化：现状报告

本文力图概述生命周期自动化的动机与目标并展示了对新技术的实际测试。虽然所描述的开发情况对新的生命周期自动化技术是一种挑战，但这种补救本身并不是最大的动力。真正驱动生命周期自动化不断改进的是我们目前只考虑了今天的开发状态的事实，明天的挑战更值得我们注意。

值得感谢的是，生命周期自动化的现实正走向我们。上面的表格显示了 Telelogic 的解决方案是怎样提供足够的能力来面对今天的挑战并展示了当今生命周期自动化的水平。它表明了这种技术正在变成现实。许多 Telelogic 的用户每天都依赖这种能力来开发系统去满足 *他们的* 客户。

更多的生命周期功能将会出现—我们是这个充满朝气的产业的一部分，能够满足开发需要的工具总是会受欢迎。在激烈的竞争中，从今天的解决方案中所获得的竞争优势将使你的事业保持优势直到生命周期自动化不在成为前沿技术的时候。Telelogic 力图成为实现这种未来理想的重要成员，也将不断创新与领导市场走向全面的生命周期自动化。

### 参考文献

- [1] Manifesto for Agile Software Development  
<http://www.agileAlliance.org>
- [2] Capability Maturity Model  
<http://www.sei.cmu.edu/cmm/cmms/cmms.html>
- [3] Royce, W. W. *Managing the Development of Large Software Systems*,  
Proceedings of IEEE WESCON, August 1970.
- [4] Barry Boehm. *Unifying Software Engineering and Systems Engineering*. IEEE Computer, March 2000.
- [5] ISO/IEC 12207, International Organization for  
Standardization/International Electrotechnical Commission, 1995.
- [6] Kent Beck. *Extreme Programming Explained: Embrace Change*.  
Addison-Wesley, 1999.
- [7] *ActiveCM* from Telelogic AB.  
<http://www.activecm.org>