第18章　需求链中的联系链

“开发工作进展如何，Glenn?”，在一次项目状态检查会上“化学制品跟踪系统”项目经理Dave问道。

“我没有按计划执行”Glenn说，“我正在应Marcie的要求添加一个销售分类查询功能，比我原先预计的工作量超期多了。”

Dave似乎有点迷惑，“好像在最近的变更控制委员会的会议中我们没有讨论过这个功能。Marcie是通过变更过程来提交要求的吗？”

“没有，她直接给了我这个建议，”Glenn说，“本该请她通过正式渠道的，但这个功能看上去较简单，所以当时我就答应她了。这个功能其实并不简单，每次当我认为该完工了，但总能意识到在另一个文件中漏了一个变更，所以不得不修改它，再测试一遍。原以为花4个小时就可以了，实际上花了4天时间，造成我没能按计划完成任务。我知道延误了工期，那我应该加上这个功能还是放弃它呢？”

一个表面上简单的软件变更往往会很复杂，花费时间很难预料。经常很难发现哪怕一个很小的修改会影响到的范围。开发过程中在同意接受建议的变更之前，要确信明白自己在做什么。

本章讲述开发过程中的需求跟踪和需求变更影响分析的相关内容。需求跟踪包括编制每个需求同系统元素之间的联系文档。这些元素包括别的需求、体系结构、其他设计部件、源代码模块、测试、帮助文件、文档等。跟踪能力信息使变更影响分析十分便利，有利于确认和评估实现某个建议的需求变更所必须的工作。

18.1　需求跟踪

跟踪能力（联系）链（traceability link）使你能跟踪一个需求使用期限的全过程，即从需求源到实现的前后生存期（Gotel and Finkelstein 1994）。第1章指出跟踪能力是优秀需求规格说明书的一个特征。为了实现可跟踪能力，必须统一地标识出每一个需求，以便能明确地进行查阅（Davis 1993）。

图18-1说明了四类需求跟踪能力链（Jarke 1998）。客户需求可向前追溯到需求，这样就能区分出开发过程中或开发结束后由于需求变更受到影响的需求。这也确保了需求规格说明书包括所有客户需求。同样，可以从需求回溯相应的客户需求，确认每个软件需求的源头。如果用使用实例的形式来描述客户需求，图18-1上半部分就是使用实例和功能性需求之间的跟踪情况。

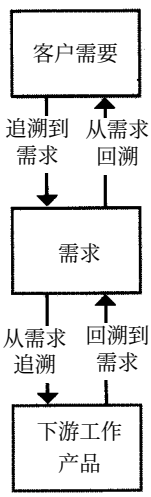

图18-1　四类需求可跟踪能力

图18-1的下半部分指出：由于开发过程中系统需求转变为软件需求、设计、编写等，所以通过定义单个需求和特定的产品元素之间的（联系）链可从需求向前追溯。这种联系链使你知道每

个需求对应的产品部件，从而确保产品部件满足每个需求。第四类联系链是从产品部件回溯到需求，使你知道每个部件存在的原因。绝大多数项目不包括与用户需求直接相关的代码，但对于开发者却要知道为什么写这一行代码。如果不能把设计元素、代码段或测试回溯到一个需求，你可能有一个“画蛇添足的程序”。然而，若这些孤立的元素表明了一个正当的功能，则说明需求规格说明书漏掉了一项需求。

跟踪能力联系链记录了单个需求之间的父层、互连、依赖的关系。当某个需求变更（被删除或修改）后，这种信息能够确保正确的变更传播，并将相应的任务作出正确的调整。图18-2说明了许多能在项目中定义的直接跟踪能力联系链。一个项目不必拥有所有种类的跟踪能力联系链，要根据具体的情况调整。

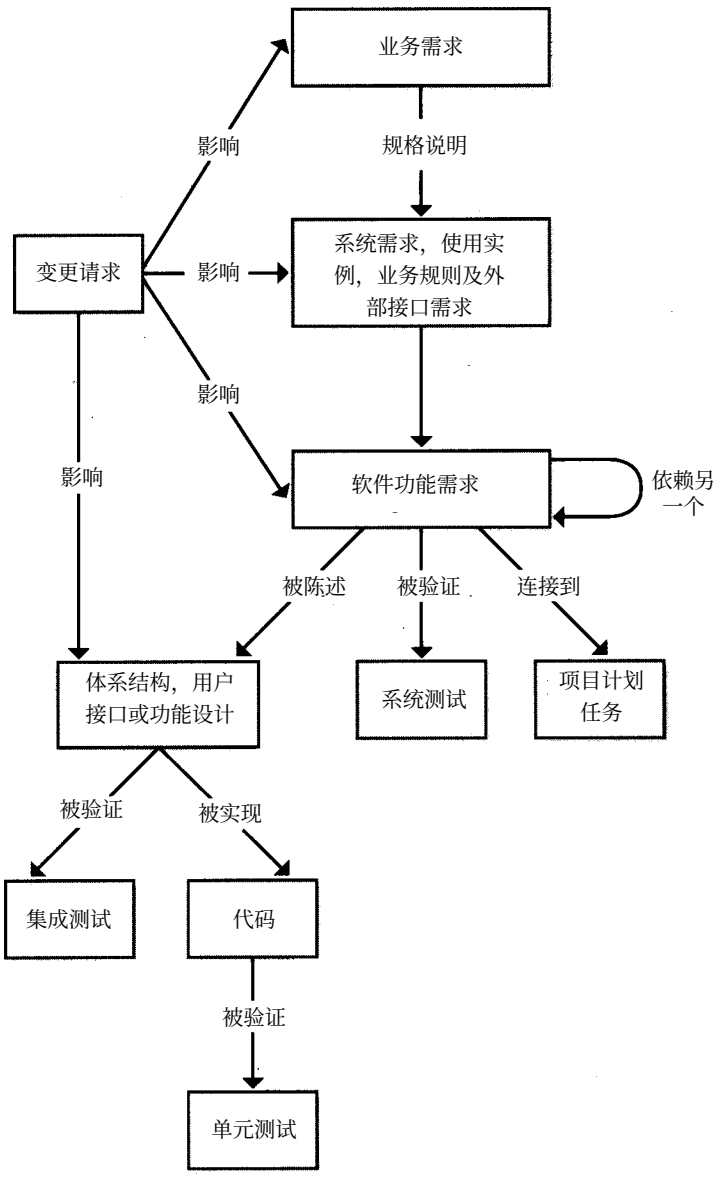

图18-2 一些可能的需求跟踪能力联系链

18.1.1 需求跟踪动机

我有一个尴尬的开发经历，其实是忽略了一个需求，所以在我自认为完成编程后，不得不返工编写额外的代码。就是因为忘了，后来不得不返工。如果忽略某几个需求造成用户不满意或发布一个不符合要求的产品，那就不仅仅是尴尬了。在某种程度上，需求跟踪提供了一个表明与合同或说明一致的方法。更进一步，需求跟踪可以改善产品质量，降低维护成本，而且很容易实现重用（Ramesh 1998）。

需求跟踪是个要求手工操作且劳动强度很大的任务，要求组织提供支持。随着系统开发的进行和维护的执行，要保持关联链信息与实际一致。跟踪能力信息一旦过时，可能再也不会重建它了。由于这些原因，应该正确使用需求跟踪能力（ Ramesh et al. 1995）。下面是在项目中使用需求跟踪能力的一些好处：

- 审核（certification） 跟踪能力信息可以帮助审核确保所有需求被应用。
- 变更影响分析 跟踪能力信息在增、删、改需求时可以确保不忽略每个受到影响的系统元素。
- 维护 可靠的跟踪能力信息使得维护时能正确、完整地实施变更，从而提高生产率。要是一下子不能为整个系统建立跟踪能力信息，一次可以只建立一部分，再逐渐增加。从系统的一部分着手建立，先列表需求，然后记录跟踪能力链，再逐渐拓展。
- 项目跟踪 在开发中，认真记录跟踪能力数据，就可以获得计划功能当前实现状态的记录。还未出现的联系链意味着没有相应的产品部件。
- 再设计（重新建造） 你可以列出传统系统中将要替换的功能，记录它们在新系统的需求和软件组件中的位置。通过定义跟踪能力信息链提供一种方法收集从一个现成系统的反向工程中所学到的方法。
- 重复利用 跟踪信息可以帮助你在新系统中对相同的功能利用旧系统相关资源。例如：功能设计、相关需求、代码、测试等。
- 减小风险 使部件互连关系文档化可减少由于一名关键成员离开项目带来的风险（Ambler 1999）。
- 测试 测试模块、需求、代码段之间的联系链可以在测试出错时指出最可能有问题的代码段。

以上所述许多是长期利益，减少了整个产品生存期费用，但同时要注意到由于积累和管理跟踪能力信息增加了开发成本。这个问题应该这样来看，把增加的费用当作一项投资，这笔投资可以使你发布令人满意同时更容易维护的产品。尽管很难计算，但这笔投资在每一次修改、扩展或代替产品时都会有所体现。如果在开发工程中收集信息，定义跟踪能力联系链一点也不难，但要在整个系统完成后再实施代价确实很大。

CMM（capability maturity model）的第三层次要求具备需求跟踪能力（ CMU/SEI 1995）。软件产品工程活动的十个关键处理领域有关于它的陈述，“ 在软件工作产品之间，维护一致性。工作产品包括软件计划，过程描述，分配需求，软件需求，软件设计，代码，测试计划，以及测试过程。” 需求跟踪过程中还定义了一些关于一个组织如何处理需求跟踪能力的期望。

18.1.2 需求跟踪能力矩阵

表示需求和别的系统元素之间的联系链的最普遍方式是使用需求跟踪能力矩阵。表 18-1

展示了这种矩阵，这是一个“化学制品跟踪系统”实例的跟踪能力矩阵的一部分。这个表说明了每个功能性需求向后连接一个特定的使用实例，向前连接一个或多个设计、代码和测试元素。设计元素可以是模型中的对象，例如数据流图、关系数据模型中的表单、或对象类。代码参考可以是类中的方法，源代码文件名、过程或函数。加上更多的列项就可以拓展到与其它工作产品的关联，例如在线帮助文档。包括越多的细节就越花时间，但同时很容易得到相关联的软件元素，在做变更影响分析和维护时就可以节省时间。

表18-1 一种需求跟踪能力矩阵

使用实例	功能需求量	设计元素	代 码	测试实例
UC-28	catalog.query.sort	class	catalog.sort()	search.7
		catalog		search.8
UC-29	catalog.query..import	class	catalog..import()	search.8
		catalog	catalog.validate()	search.13
				search.14

跟踪能力联系链可以定义各种系统元素类型间的一对一，一对多，多对多关系。表 18-1允许在一个表单元中填入几个元素来实现这些特征。这里是一些可能的分类：

- 一对一 一个代码模块应用一个设计元素。
- 一对多 多个测试实例验证一个功能需求。
- 多对多 每个使用实例导致多个功能性需求，而一些功能性需求常拥有几个使用实例。

手工创建需求跟踪能力矩阵是一个应该养成的习惯，即使对小项目也很有效。一旦确立使用实例基准，就准备在矩阵中添加每个使用实例演化成的功能性需求。随着软件设计、构造、测试开发的进展不断更新矩阵。例如，在实现某一功能需求后，你可以更新它在矩阵中的设计和代码单元，将需求状态设置为“已完成”。

表示跟踪能力信息的另一个方法是通过矩阵的集合，矩阵定义了系统元素对间的联系链。例如：

- 一类需求与另一类需求之间。
- 同类中不同的需求之间。
- 一类需求与测试实例之间。

可以使用这些矩阵定义需求间可能的不同联系，例如：指定 /被指定、依赖于、衍生为以及限制/被限制（Sommerville and Sawyer 1997）。

表18-2说明了两维的跟踪能力矩阵。矩阵中绝大多数的单元是空的。每个单元指示相对应行与列之间的联系，可以使用不同的符号明确表示“追溯到”和“从……回溯”或其他联系。表18-2使用一个箭头表示一个功能性需求是从一个使用实例追溯来的。这些矩阵相对于表18-1中的单跟踪能力表更容易被机器自动支持。

表18-2 反映使用实例与功能需求之间联系的需求跟踪能力矩阵

功能需求	使用实例			
	UC-1	UC-2	UC-3	UC-4
FR-1 FR-2 FR-3				

(续)

功能需求	使用实例			
	UC-1	UC-2	UC-3	UC-4
FR-4 FR-5 FR-6				

跟踪能力联系链无论谁有合适的信息都可以定义。表 18-3定义了一些典型的知识源，即关于不同种类源和目标对象间的联系链。定义了可以为工程项目提供每种跟踪能力信息的角色和个人。

表18-3 跟踪能力联系链可能的信息源

链的源对象种类	链的目的对象种类	信 息 源
系统需求	软件需求	系统工程师
使用实例	功能性需求	需求分析员
功能性需求	功能性需求	需求分析员
功能性需求	软件体系结构元素	软件体系结构（设计）者
功能性需求	其他设计元素	开发者
设计元素	代码	开发者
功能性需求	测试实例	测试工程师

18.1.3 需求跟踪能力工具

由于联系链源于开发组成员的头脑中，所以需求跟踪能力不能完全自动化。然而，一旦已确定联系链，特定工具就能帮你管理巨大的跟踪能力信息。可以使用电子数据表来维护几百个需求的矩阵，但更大的系统需要更“鲁棒”的解决办法。

第19章描述了具有强大需求跟踪能力的商业需求管理工具。这些工具均使用如表 18-2的跟踪能力矩阵。可以在工具的数据库中存储需求和其他信息，定义不同对象间的联系链，甚至包括同类需求的对等联系链。有一些工具需要区分“追溯到（跟踪进）”与“从……回溯（跟踪出）”关系，自动定义相对的联系链。这就是说，如果你指出需求 R追溯到测试实例T，工具会自动定义相对的联系“ T从R回溯”。还有一些工具可以在联系链某端变更后将另一端标为“可疑”。可以让你检查确保知道变更的后续效果。

这些工具允许定义“跨项目”或“跨子系统”的联系链。我了解到一个有 20个子系统的大项目，某些高层产品需求建立在多个子系统之上。有些情况下，分配给一个子系统的需求，实际上是由另一个子系统提供的服务完成的。这样的项目采用商业需求管理工具可以成功地跟踪这些复杂的跟踪能力关系。

18.1.4 需求跟踪能力过程

当你应用需求跟踪能力来管理工程时，可以考虑下列步骤：

1) 决定定义哪几种联系链，可以参见图 18-2来进行。
2) 选择使用的跟踪能力矩阵的种类，是表 18-1还是表18-2。
3) 确定对产品哪部分维护跟踪能力信息。由关键的核心功能、高风险部分或将来维护量

大的部分开始做起。

4) 通过修订过程和核对表来提醒开发者在需求完成或变更时更新联系链。

5) 制定标记性的规范，用以统一标识所有的系统元素，达到可以相互联系的目的（ Song et al. 1998）。若必要，作文字记录，这样就可以分析系统文件，便于重建或更新跟踪能力矩阵。

6) 确定提供每类联系链信息的个人。

7) 培训项目组成员，使其接受需求跟踪能力的概念和了解重要性、这次活动的目的、跟踪能力数据存储位置、定义联系链的技术 —— 例如，使用需求管理工具的特点。确保与会人员明白担负的责任。

8) 一旦有人完成某项任务就要马上更新跟踪能力数据，即要立刻通知相关人员更新需求链上的联系链。

9) 在开发过程中周期性地更新数据，以使跟踪信息与实际相符。要是发现跟踪能力数据没完成或不正确那就说明没有达到效果。

18.1.5 需求跟踪能力可行吗，必要吗？

你可能会认为建造一个需求跟踪能力矩阵与它的价值相比麻烦多过益处，或者虽然简单但却不实用。看一下这个例子：某个会议的参加者在飞机制造厂工作，有一次他告诉我他所在公司最新型的喷气客机用户需求说明书足有一人高，而且有完整的需求跟踪能力矩阵。我非常高兴听到开发人员如此严格地管理他们的软件需求。对有很多子系统的巨大产品进行跟踪能力管理是一项巨大的工程，但他们知道这很必要。

并不是所有的公司都会因为软件问题而造成严重的结果，然而应该严肃地对待需求跟踪，尤其对涉及你业务核心的信息系统。有一次当我在一个专题会上描述完跟踪能力后，在场的一个大公司的CEO问道：“ 为什么不对战略商业系统作一个跟踪能力矩阵？ ” 这是一个很好的问题。考虑了应用技术的成本和不使用的风险后，才能决定是否使用任何改进的需求工程实践。随着软件的发展，要把时间投向回报丰厚的地方。

18.2 变更需求代价：影响分析

许多开发人员有过类似本章正文前的那种对话情况。一个表面上很简单的变更转变成很复杂的局面。只要允许需求变更或添加新特性，这种情况就免不了。开发人员往往对建议的软件变更成本或其它衍生结果不 —— 或不能 —— 提供出准确的评估。“ 变更是免费的 ” 这种误解是造成项目范围延伸的一个原因。人们往往只有在知道变更的成本后才能做出理智的选择。

影响分析是需求管理的一个重要组成部分（ Arnold and Bohner 1998）。影响分析可以提供对建议的变更的准确理解，帮助做出信息量充分的变更批准决策。通过对变更内容的检验，确定对现有的系统做出是修改或抛弃的决定，或者创建新系统以及评估每个任务的工作量。进行影响分析的能力依赖于跟踪能力数据的质量和完整性。

没有人愿意做一个费时费力还要担心意想不到情况的需求变更。在职业生涯中，绝大多数开发人员会遇到要求添加 “ 没有代价且不影响进度的变更 ” 的要求。对这样令人奇怪的要求的正确回答是 “ 不行，” 变更只能在项目时间、预算、资源的限制内进行协商。

18.2.1 影响分析过程

项目变更控制委员会通常会请资深开发人员对提出的需求变更申请进行影响分析。为了帮助影响分析员理解接受一个建议变更的影响，可设计一系列问题核对表，如图 18-3所示。第二个核对表如图18-4，用来帮助确定涉及的软件元素。熟练以后，可以按照具体情况修改。

- 基准（线）中是否已有需求与建议的变更相冲突？
- 是否有待解决的需求变更与已建议的变更相冲突？
- 不采纳变更会有什么业务或技术上的后果？
- 进行建议的变更会有什么样的负面效应或风险？
- 建议的变更是否会不利于需求实现或其它质量属性？
- 从技术条件和员工技能的角度看该变更是否可行？
- 若执行变更是否会在开发、测试和许多其它环境方面提出不合理要求？
- 实现或测试变更是否有额外的工具要求？
- 在项目计划中，建议的变更如何影响任务的执行顺序、依赖性、工作量或进度？
- 评审变更是否要求原型法或别的用户提供意见？
- 采纳变更要求后，浪费了多少以前曾做的工作？
- 建议的变更是否导致产品单元成本增加？例如增加了第三方产品使用许可证的费用。
- 变更是否影响任何市场营销、制造、培训或用户支持计划？

图18-3 建议的变更涉及的问题核对表

- 确认任何用户接口要求的变更、添加或删除。
- 确认报告、数据库或文件中任何要求的变更，添加或删除。
- 确认必须创建、修改或删除的设计部件。
- 确认源代码文件中任何要求的变更。
- 确认文件或过程中任何要求的变更。
- 确认必须修改或删除的已有的单元、集成或系统测试用例。
- 评估要求的新单元、综合和系统测试实例个数。
- 确认任何必须创建或修改的帮助文件、培训素材或用户文档。
- 确认变更影响的应用、库或硬件部件。
- 确认须购买的第三方软件。
- 确认在软件项目管理计划、质量保证计划和配置管理计划等中变更将产生的影响。
- 确认在修改后必须再次检查的工作产品。

图18-4 变更影响的软件元素核对表

下面是一个评估需求变更影响的简单例子。许多评估问题的出现是因为评估者没有完全按此行事。所以，这个影响分析方法强调广泛的任务确认。对于重大的变更，小组 —— 不只是一个开发者 —— 应该做影响分析和工作量估算来确保不忽略重要的任务。

1) 按图18-3进行一遍。

2) 按图18-4进行一遍，要使用有效的跟踪能力信息。有一些需求管理工具包含影响分析报告并且能发现受变更影响的系统元素。

3) 使用如图18-5所示的一张工单（worksheet）来评估预期任务要求的工作量，绝大多数的变更仅要求工单所列任务的一部分。

4) 求评估工作值的总和。

5) 确认任务执行的顺序，这些任务如何同当前的计划任务配合？

6) 决定变更是否处于项目的临界路径。如果一个处于关键路径的任务延期，项目的完成之日将遥遥无期。每个变更都会消耗资源，如果能避免变更影响关键任务，则变更不会造成整个项目延期。

7) 估计变更如何影响进度和费用。

8) 通过与其它任意需求的收益、代价、成本和技术风险的比较来评估变更的优先级（有关详情请见第13章）。

9) 向变更控制委员会报告影响分析结果，他们可以在采纳或拒绝变更的决策过程中使用这些评估信息。

在绝大多数实例中，完成这些过程不会超过一到二小时。对于一个繁忙的开发人员来说似乎浪费了很多时间，其实为了确保开发人员对有限的资源的精打细算，这只是一笔很小的投资。如果不用系统的评估技能从容地评估出变更影响，就去做吧；但要小心别让自己进入“流沙区”。

工作量（劳动时数）	任　务
_____	更新软件需求规格说明书或需求数据库
_____	开发并评估原型
_____	创建新的设计部件
_____	修改已有的设计部件
_____	开发新的用户界面部件
_____	修改已有的用户界面部件
_____	开发新的用户文档和帮助文件
_____	修改已有的用户文档和帮助文件
_____	开发新的源代码
_____	修改已有的源代码
_____	购买和集成第三方软件
_____	修改构造文件
_____	开发新单元测试和综合测试
_____	进行单元测试和综合测试
_____	写新的系统测试实例
_____	修改已有的系统测试实例
_____	修改自动测试驱动程序
_____	进行回归测试
_____	开发新报告
_____	修改已有的报告
_____	开发新的数据库元素
_____	修改已有的数据库元素
_____	开发新的数据文件
_____	修改已有的数据文件
_____	修改各种项目计划
_____	更新别的文档
_____	更新需求跟踪能力矩阵
_____	检查工作产品
_____	根据测试和检查情况返工
_____	总计劳动时数

图18-5　评估需求变更的劳动时数

18.2.2 影响分析报告模板

图18-6是一个模板，可以用来报告进行需求变更的影响分析。使用模板可以帮助变更控制委员会找到有用的信息来作出正式的决策。实现此项变更的开发人员可能需要详细的分析情况和工作量计划工单，但变更控制委员会仅需要影响分析的总结。可以视情况调整模板。

变更请求ID_____
标题_____
描述_____
分析者_____
日期_____
优先权评估：
相关收益_____（1-9）
相关代价_____（1-9）
相关成本_____（1-9）
相关风险_____（1-9）
最终优先级_____
预计总耗时_____劳动时数
预计损时_____劳动时数
预计对进度的影响_____天数
额外的成本影响 _____金额
质量影响_____
_____
被影响的其他需求_____
被影响的其他任务_____
要更新的计划 _____
综合的事项 _____
生存期成本事项 _____
可能的变更所需检查的其他部件 _____
_____

图18-6 影响分析报告模板

下一步：

- 为你当前开发的系统最核心部件建立一个 15~20需求的跟踪能力矩阵。尝试一下表18-1和18-2的方法。随着项目的开发逐步扩展矩阵。评估一下什么方法最有效以及什么样的收集和存储跟踪能力信息的方法最适合你。
- 下一回评估一个需求变更请求，首先用旧方法来预算耗时，再用本章的影响分析方法计算一下。当最终实现了这个变更，再比较一下两个方法哪个估算更准确。随着经验的积累可以修改核对表和工单。
- 当你有机会对一个文档不全的项目做维护时，应该用反向工程的分析方法来处理要修改的部分，并且把经念教训记录下来。对你所处理的难题建立一部分跟踪能力矩阵，以便以后在其基础之上工作的人有据可查。当你的工作小组继续维护该产品时，进一步扩充跟踪能力矩阵。