

进度管理：需求迭代与项目风险控制 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/269/2021_2022__E8_BF_9B_E5_BA_A6_E7_AE_A1_E7_c67_269582.htm 篇文章由舜亚科技的Jimmy发表在《程序员》2007.2期，其中的案例全部引自本公司的项目。 作者介绍：柯自聪/eamoi 舜亚科技软件工程师，专注于Web应用程序开发，关注OA、门户、电子政务、电子商务领域、RIA，著有《Ajax开发精要--概念、案例与框架》一书以及《Ajax开发简略》、《Liferay Portal二次开发指南》等开源文档。 全文：软件项目是需求驱动的典型代表，项目从立项、开发、测试到交付，需求的变化迭代是很正常的事情，这点对于大型项目尤其明显。需求迭代如果控制不好，很容易增大项目的风险，导致项目的失败。与国内的很多软件公司相似，笔者所参与的项目也存在需求迭代的问题。 本文从需求迭代入手，结合项目实际，探讨需求迭代与项目风险控制的关系，希望项目需求有序迭代。 需求迭代，不可避免的轮回 软件项目的启动源于市场和客户的需求，通过对市场的需求调查以及典型目标客户的需求访问抽象出需求规格说明书，进而才开始原型系统的设计，经过对原型系统的评估之后启动真实系统的设计和开发。 在原型系统设计阶段，由于各种各样的因素，比如客户没有将实际需求表达清楚，或者需求分析人员对业务的理解有偏差，据此设计出来的原型系统可能与市场、客户的真实需求不是很匹配，那么随着原型系统构建的深入，必然触发需求的迭代。 在真实系统的设计和开发过程中，随着对系统的理解的深入，客户也可能对需求进行深化、扩展或者变更，研发工程师对需求的消

化，这也会触发需求的迭代。即使真实系统交付使用，随着业务的发展，客户的需求可能发生变化；而且客户在使用系统的过程中，必然会对系统提出进一步改进的要求，修改原有功能的操作方式，增加新的功能，这些也会要求需求的进一步迭代。在这一系列的迭代过程中，如果没有很好的控制迭代的过程，评估迭代的成本，有效管理迭代的需求，那么很容易形成需求迭代无穷无尽的假象，项目团队穷于应付每一次需求迭代，项目开发高度紧张，发布日期遥遥无期，这样必然给项目带来很高的风险。Diapers项目是一个面向北美市场的电子商务站点，已经运行三年。最近客户决定对Diapers项目进行升级改造。项目经理或者需求分析工程师负责沟通客户，分析抽象客户的真实需求，并撰写需求说明书；软件工程师根据需求说明书拟定技术方案，并着手进行编码；测试工程师根据需求说明书和测试用例对项目进行测试；项目经理引导客户确认项目的最终功能呈现，并在必要的时候启动需求迭代过程。由于Diapers是来自北美的外包项目，双方的沟通存在时间差，项目团队也没有条件与客户面对面的沟通。在整个项目的升级改造过程中，由于业务理解的偏差以及沟通不畅，需求经过了多次迭代；需求每迭代一次，团队成员都需要面对一堆冗长的需求说明书。由于Diapers已经是正式运营的站点，客户来自市场的压力同时也转嫁到项目团队身上，项目发布的压力一直困扰着团队成员。从Diapers项目的进展来看，需求的迭代似乎就是无穷无尽的轮回。主动触发需求迭代，给予足够的消化时间导致Diapers项目的现状的主要原因是被动的进行需求迭代，迭代被动的由客户的反馈触发。每次需求迭代都可能打乱团队

的开发计划，影响项目的发布，给团队带来更大的发布压力。因此，必须想方设法掌握需求迭代的主动权。针对每次需求迭代给予充分的消化时间是一种有效的方式。从Diapers项目的情况来看，上一次需求还没有消化处理完毕，新的需求迭代又要开始了。项目发布迭代的速度根本就跟不上需求迭代的速度，新的需求一直步步进逼。在这种情况下，测试工程师压根儿就没有时间对项目进行全面的足够的测试。找到问题的本质，Diapers项目团队开始调整发布节奏，加大人力资源投入，加快消化需求的速度；针对沟通不足的问题，项目经理集中精力与客户沟通，在双方时间交叉的部分尽量把有疑问的需求沟通清楚；发布节奏调整后，客户就有时间与项目团队同步开展测试工作，bug也能够在第一时间处理。调整后，项目团队有足够的时间来消化每次迭代的需求，也有足够的时间对项目进行测试。尽早发布原型系统是主动触发需求迭代的另一种有效方式。原型系统通常快速构建，着重在界面的呈现和功能的模拟，通过虚拟数据模拟真实系统的运行情况。其能够在很大程度上模拟未来真实系统的呈现，在短时间内将抽象的客户需求表现出来，作为和客户进行沟通的有效媒介。相对于一堆抽象的文档，使用原型系统，客户更容易尽早发现真实系统与他们的需求之间的差距，减少未来需求迭代的次数。因此，在需求抽象过程中，应该通过原型系统作为双方沟通的桥梁和媒介，双方应该先就原型系统的呈现展开讨论。另外，原型系统的发布时间也是比较重要的，在项目启动后应该尽早发布原型系统。Claim项目则是一个商业意外险理赔平台，为北美客户提供商业意外险的在线申报、理赔服务。在项目启动的初期，项目团队在理解抽

象需求的基础上，第一时间发布了原型系统，使用虚拟数据模拟真实系统的界面呈现。这个项目比较有利的是，客户自己聘请了需求分析人员，能够最大程度的理解业务需求，正确的表述客户的需求，并绘制详细的原型界面；这点在双方的沟通和系统开发过程中发挥了比较显著的作用。由于Claim项目的需求迭代节奏一直在项目团队的可接受范围，所以项目一直有条不紊的推进，虽然需求也经过了多次迭代，但终究还在可控的范围内。评估每一次迭代的成本和风险能够预见的是，需求的每次迭代都会不同程度的对项目产生影响，对此需要评估由此所带来的成本。不只是项目经理和需求分析工程师，软件工程师和测试工程师也应该参与这个过程，评估此次迭代是否会影响现有的技术架构，哪些功能点可能受到影响，哪些系统模块需要修改，测试用例是否应该重新编写，团队需要为此次迭代额外付出多少时间成本，是否会造成项目的延期等等。评估之后，如果需求迭代对项目的进度可能造成比较明显的影响，项目经理应该和客户有效沟通，告知需求迭代的成本，尤其是时间成本。另外，需求的每次迭代也必然给项目带来一定的风险，包括技术风险和发布风险。迭代后的需求可能影响原有的技术方案，尤其是核心业务逻辑的变更。团队要重新对技术方案进行梳理，检查该技术方案是否仍然可以达到既定的目的。需求迭代之后，软件工程师需要一定的时间调整开发进度，测试工程师也需要根据新的需求对系统重新测试，这必然影响项目的发布周期；作为项目经理，应该预见到这一点。GS项目是某公司重点研发的一个以政府机关行政审批业务为服务目标的产品，在其进行产品升级改造的同时，其竞争对手也在着手准备同

类产品的新版本发布，市场的压力要求产品尽快完成版本的升级。但是在新产品即将进入集成测试阶段的时候，公司突然决定对产品的界面进行比较重大的调整。这一次调整导致代码和测试的返工，使该产品的发布时间推迟了两个月，错过了销售的黄金期，市场和客户对于新产品的期待已经逐渐降低，结果产品的销售量远远不如预期。如果公司之前对界面需求迭代有比较清晰的成本和风险评估，那么应该不会这么仓促的触发迭代。Diapers项目团队意识到Diapers项目的需求迭代的周期是比较短的，因此对于每次迭代的需求，软件工程师和测试工程师都会协同项目经理进行评估，判断消化所有需求并测试所需要投入的工作量，以及由此所可能带来的时间成本和技术风险，团队成员已经彻底摆脱了害怕需求迭代的心态。

明确项目发布的需求边界 软件不是十全十美的，需求的迭代是永无止境的。需求的迭代周期是不定的，与其在最终版本中包括所有的需求，不妨在这期间发布若干个小版本，明确每个小版本的需求边界。这好比长跑途中的若干个里程碑，每跨过一个里程碑就意味着向重点又前进了一步。每个小版本都包含有限的功能需求，测试工程师可以针对这些功能需求同步展开测试工作，提早触发Bug，尽量争取测试时间。客户也可以从这些小版本中提前看到真实系统的雏形；随着版本的逐步升级，项目距离发布日期也越来越近，和需求的差距也越来越小。工欲善其事，必先利其器。我们可以利用一些现成的工具来管理需求边界和跟踪Bug，比如JIRA。JIRA是集项目计划、任务分配、需求管理、错误跟踪于一体的商业软件，其提供了问题跟踪管理、问题跟进情况的分析报告、项目类别管理、组件/模块负责人、项目email

地址等功能。许多著名的开源项目都采用了JIRA。通过JIRA，可以整合客户、项目经理、开发人员、测试人员，使各种角色各司其职，团队内部信息能够很快得到交流和反馈，潜在的问题提前暴露，促进项目的可控。JIRA以工作流的思想融合了项目管理、任务管理和缺陷管理等思维，允许设定项目的模块和版本，并为每个需求设置预期日期，将任务的处理指定到人。JIRA允许为每个项目设置优先级，比如Blocker、Critical、Major、Minor、Trivial，标识每个任务的重要程度。如果任务定义了优先级，那么在每个人的桌面上，任务会自动排列。这点对于多任务的项目尤其重要。预见到需求迭代的被动性后，Diapers项目团队在Diapers项目上全面启动了JIRA进行项目管理，将需求分解细化后进入JIRA，排定任务的优先级并指定到人，确定每次小版本发布的需求编号，不定期的发布小版本。结合SVN等版本控制工具，Diapers项目团队能够将功能需求迭代的粒度控制到最小，项目逐步推进，客户对项目的进度有充分的了解，项目经理也能够准确把握项目的进度，团队中充满了乐观的情绪。安抚团队成员的情绪 工程师对于冗长的需求说明书有天生的恐惧感，开发周期拉得太长，似乎需求迭代无穷无尽。如果需求的迭代周期不在可控范围之内，项目的发布边界模糊不清，项目发布的日期自然也遥遥无期。由此带来的结果是团队每天紧赶慢赶的跟踪需求迭代，消化处理新的需求，工作气氛也是高度紧张。每一次需求迭代，都会进一步增加这种紧张情绪。项目经理应该把握项目的进展情况以及客户的真实需求，也要知悉客户的需求底线，更要在必要的时候安抚团队成员的情绪。当原始需求第一次被抽象出来的时候，团队的第一要务

是快速构建原型系统作为和客户沟通的主要媒介和依据，项目经理应该引导团队把握这一点。之后的每一次需求迭代，项目经理要将需求分解细化，控制需求的粒度，并且确定优先级，消除团队成员的焦急情绪，按照先后顺序逐步的处理每一个粒度的需求，以发布每阶段的小版本为阶段性的目标。在这个过程中，需求细化到最小粒度，还需要注意到每个需求之间的关联关系，关联的需求要优先集中处理，是降低每个小版本之间的耦合度。Diapers项目自从将需求细化成一个个任务并进入JIRA控制之后，软件工程师每天的只需要按照顺序处理JIRA上面的任务，并及时将代码以最小粒度的形式通过SVN工具提交；测试人员根据发布边界所指定的版本号从SVN下载最新代码测试，确认并关闭相应的任务；项目经理引导团队成员遵循规范的需求迭代程序，有条不紊的处理需求，轻松应对需求迭代。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com