

案例：无清晰策略的快速开发

张辉准备领导他在公司的第 2 个项目。他的前一个项目进行得并不如人意。原计划要求他的项目小组 12 个月内交付 CRM 2.0 版，但实际花上了 18 个月。项目组在项目开始时就已经认识到项目计划进度的紧迫性，因而项目一开始就处于“死亡进程”（Death March）之中。经过整整 18 个月每天 12 小时，每周 6~7 天的艰苦挣扎，到项目结束时，6 个小组成员中有 2 个退出，小组中最具实力的开发人员文志从上海骑自行车出走，目的不明。他从北京的长城脚下发给张辉一张他骑着自行车的照片，文志说他不是退出，但没人知道他什么时候能回来。

CRM 3.0 版要在 CRM 2.0 版上市的 12 个月后发布，项目收尾、总结、休假已经 2 个月了。张辉准备开始新的尝试。离交付 CRM 3.0 版还有 10 个月的时间。他约了上司陈键与他讨论项目计划，陈键可以帮助他为开发人员提供尽可能的工作条件。用户文档负责人和测试负责人也一同参加了讨论。

陈键说：“3.0 版必须超过竞争对手，所以我们在这个项目上需要投入更大的力量。我想你们可能还没意识到。从上个版本开始我们公司就已经落后了，这次公司准备全力以赴支持这个项目。我已经批准为项目组每个成员提供私人办公室，最新型的计算机以及免费的咖啡。你觉得怎样？”

张辉回答：“听起来很有吸引力，不过 除此之外，由于所有开发人员都是有经验的，我想主要应该多一些激励和支持措施，并使之从中获益。我不想事无巨细地管理他们，我想让每个开发人员都对系统的某一部分负责。上次，我们已经碰到许多接口问题，所以，这次我想花一些时间设计不同模块之间的接口，然后放手让他们自己去干。”

用户文档负责人说：“如果这是一个 10 个月的项目，我们需要在第 8 个月的时候锁定软件，及时准备用户文档。上次，直到项目结束，开发人员还在进行修改，令人尴尬的 readme 文件有 20 页长。用户手册在审查的时候总是被枪毙，如果你能做到可视化锁定，你的开发方法听起来就有道理了。”

测试负责人补充说：“我们也需要在可视化锁定的同时写出自动测试脚本。”

张辉赞同可视化锁定方法。陈键随即批准了张辉的总体步骤，并告诉他要保持大家步调一致。

当项目开始时，开发人员对他们的私人办公室，新的计算机及咖啡很满意，他们干劲十足地开始了工作，并主动工作到深夜。

时间一个月一个月地过去，项目进展平稳，他们已经完成了早期的软件原型，继续进行编码设计。管理层也一直在监视项目的进展，每件事情似乎都在顺利进行。

项目进行到了第 4 个月的时候，文志结束了他的自行车旅行，重新回到了项目组，并带回了一些骑车旅行时产生的新想法。张辉担心文志能否在允许的时间内完成那么多任务，但文志承诺，无论有多大的工作量，一定按时完成任务。

项目组成员各自独立做着自己的工作，而随着可视化锁定日期的来临，他们开始进行代码集成。他们在可视化锁定最终截止日期前一天的下午 2 点开始工作，但很快发现程序不能链接(link)，更不用说运行了。代码在链接时有十个语法错误，而且每处理一个错误就会产生 10 个以上的新错误。他们直到午夜也没结果，只好决定第 2 天再说。

第二天早晨，陈键约见项目组成员：“程序移交给文档和测试人员了吗？”

张辉说：“还不行，我们还有些集成问题，可能今天下午就可以移交了。”项目组又工作了整个下午和晚上，但还是没能解决已经发现的所有问题，最后，他们只得承认无法预知集成需要多长时间。

他们整整化了 2 周时间处理这些语法错误，才得以使系统能够运行，当项目组比计划推迟了 2 周将阶段定版的系统移交时，测试和文档人员迅速做出反应，拒绝接收这样的版本。文档负责人说：“系统太不稳定，每隔几分钟就会崩溃，同时，有太多的系统漏洞，不能编写文档。”

测试负责人也赞同道：“系统太不稳定，你每次选择一次菜单，系统就会瘫掉，测试人员根本无法完成测试报告。”

张辉同意他们的说法，并表示他的项目成员会全力解决这些问题。陈键提醒他们注意 10 个月的最后截止日期，并说这个产品不能再像上个产品那样延期了。

他们又花了一个月的时间解决产品稳定性的问题，以使文档和测试工作能够进行，那时，距离 10 个月的最后截止日期只有 2 周时间了，他们只能更加拼命地工作。

但测试发现问题的速度远比开发人员解决问题的速度快，处理系统这部分的错误经常会导致系统其他部分的问题。已经无法按预计的 10 个月交付产品了。陈键召开了一个紧急会议，他说：“我看各位工作都很辛苦，但结果不够好，我已经为你们提供了各种尽可能的支持，但我并没有看到最终的软件，如果你们不能迅速完成产品，公司将会倒闭。”

伴随着巨大的压力，士气渐落。数月过去了，产品开始稳定了，但陈键仍对项目组施加压力，有些工作的效率极其低下。

尽管文志也在不停地工作，但他还是比项目中的其他人交付软件的时间晚。他的代码没有错误，但他对用户界面的控件做了些修改，导致测试脚本和用户文档与之不匹配。

张辉约见了测试和文档负责人：“你们可能不高兴，但我们只能有两种选择，要么保持文志的代码不变，重新进行测试并重新编写用户文档，要么仍掉文志的代码，将其彻底重写。文志不愿意重写他的代码，这个项目组也没人能做这件事了。这样看来只能请你们修改用户文档和测试用例了。”经过一翻争执，测试和文档负责人勉强同意了张辉的要求。

到项目结束时，开发这个软件整整花了 15 个月的时间，由于以上的变化，用户文档的印刷错过了计划安排的最佳时间点，公司在开发人员完成刻盘工作后，不得不又为印刷用户文档等候了 2 周，才将产品运出公司，投向市场。产品发布后，用户对 CRM 3.0 版反映冷淡，几个月时间，其市场份额就从第 2 位降到了第 4 位。张辉意识到他的第 2 个项目像第 1 个项目一样，又失败了。

讨论：本案例错在哪里？请列出所有 bad practices.

案例：具有清晰策略的快速开发

仍是 CRM3.0 项目，现在由张莉承担项目经理。

在项目组的第一次会议上，她介绍了项目组的成员，而后切入主题，“我在整个公司范围内收集了其他项目的总结报告。”她说：“我已经列出足有 1 英里长的公司其他项目在运作中发生的所有错误，我将把这个单子贴到会议室里，我想在我们开始犯其中某个错误时，就在上面画一个标记，如果你们还知道上面没有列出的其他错误或我们可能会犯的任何潜在的错误，请加在上面，我们不想重蹈覆辙。”

“选择你们参与本项目是因为你们每位都有开发技术基础，我想你们一定知道做好需求分析和设计以使我们能够不必返工而浪费时间意味着什么，所以，我要求项目组的每位员工要用心工作，而不是辛苦工作，工作太辛苦会导致出现更多的错误，我们没有时间处理这些错误。”

“我也同时制定了配套的风险管理计划，我们面对的是一个具有挑战性的进度计划，所以我们不能让能够防止的风险发生。我们现在面对的最大风险是计划的进度无法实现，我想我们应该在本周末在对进度计划进行一次评估，如果认为进度计划无法实现，我们将讨论提出一份更现实的计划。”

每位项目成员都点头，对经过“死亡进程”项目的人员来说，张莉的讲话让他们长出了一口气。

1 周后，张莉约见了她的上司陈健：“项目组已经很仔细地审视了项目计划，陈健，得出的结论是，我们只有 5% 的机会在项目截止日期完成当前定义的功能，这是假设没有任何改变的情况下，当然，有些事情总是要改变的。”

“真是糟透了。”陈健说。“我们至少需要有 50% 的机会按时交付软件，并且我们还要能够针对今后 10 个月市场的变化随时修改项目要求，对此，你有什么建议？”

“我们还没有完成对产品定型，所以，我们还是有一定灵活度的。”张莉说。“但我认为即使根据目前的需求分析，我们也需要花 8~30 个月时间，我知道这个时间跨度是比较大，但这也对我们在产品完全定型之前的工作很有利。我们需要 10 个月后提供产品，对吗？我认为，可以考虑再加入一些开发人员，然后建立一个渐进式的交付计划，今后我们每 2 个月推出一个交付版本，第一个交付版本决定在第 6 个月末完成。”

“听起来不错。”陈健说：“除此之外，我想对这个项目而言，功能比进度更重要，我会在找一些人员谈谈，然后我们再定。”

当陈健再找张莉时，他告诉她公司愿意将软件计划时间延长到 12 个月，并还是希望实现所要求的功能，另外，为安全起见，可以采用渐进式软件交付计划。张莉轻松了许多，并说她认为这是一个很现实的目标。

项目经过初期的几周后，她的项目组已经建立了详细的用户界面原型，并与潜在客户进

行交流，根据客户回馈对原型进行了几次修正。

张莉继续维护她的风险列表，并确定了这个项目的 3 个主要风险：1) 可能导致大量重复工作和进度延期的质量低下风险；2) 具有挑战性的进度本身的风险；3) 因市场竞争，要求软件功能不断增加而导致的竞争功能风险。张莉感到质量低下风险可以通过渐进交付计划来消除。他们将在第 6 个月时将第 1 个交付版本送交到测试部门，他们会按用例进行测试。

进度计划风险可以通过产品功能的优先级次序由项目组自己来消除。他们在 12 个月内会尽可能开发更多的功能，而通过每 2 个月交付 1 个版本，他们可以确保在需要的时候有东西可以交付。

项目组采用两种方式消除竞争功能风险。他们花了大约 3 个月的时间开发设计方案，该方案包含了所有已定义原型的功能和其他一些他们认为应该包含在 3.0 版中的功能的框架。这种设计使系统更容易适应可能产生的各种改变，同时他们还分配出时间在第 10 个月分析竞争对手的产品，修改原型，并在最后的 2 个月中实现必要的竞争功能。

在第 4 个月时，随着设计方案的完成，项目组制定了一组细化的里程碑，制定了到第 6 个月时，发布第 1 个可交付测试版本所遵循的原则与途径。第 6 个月交付的版本并不完善，但应该保证质量，这可以为下一步工作打下坚实的基础。在项目组顺利交付第 6 个月的版本后，项目组又为第 8 个月交付制定细化的里程碑，并使用同样的方法到达了第 10 个月的里程碑处。

在第 10 个月结束时，项目组按计划对竞争对手产品进行研究。竞争对手已经在第 8 个月发布了一个好产品，它已经包含了一些 CRM3.0 出于竞争目的而需要包含的功能。项目组立刻将这些新功能加入到优先队列中，重新分配优先次序，并制定了最后 2 个月的细化的里程碑。

几乎同时，蒋华，一个初级开发人员发现了一种可以将产品对话框进行更好的组织的方法，并建议提交到项目组成员碰头会上。会上，高级开发人员唐国刚对此做出的反应是：“这是一个非常好的想法，我认为我们应该采用这一方法修改我们的设计，但不是现在，蒋华，对你来讲，进行这样的改变只是一天的工作量，但会影响文档编制进度达一周或更多时间，将其放到 4.0 版本中如何？”

蒋华说：“我没想到对文档进度有这么大的影响。这是个好的方法，我想请求在未来修改设计时采用。”

在达到第 12 个月的里程碑时，项目组按计划交付了终版软件。由于从第 6 个月开始测试时 CRM 的质量就是优秀的，所以，文档在等待正式版软件交付期间，已经可以基于详细的用户界面进行编写了。文档与软件同步准备就绪。开发人员没有实现一些低优先级的功能，但他们实现了所有的重要功能，CRM3.0 是成功的。