RUP/XP 指南：结对编程

Robert C. Martin
Object Mentor, Inc.

Rational Software 白皮书

TP 158, 3/01

Rational®
the software development company

目录

概述

结对简要描述

结对编程是由多对程序员编写一个项目软件的技术。每对人员在一台工作站上一起工作。其中一个成员驱动工作站，另一个成员仔细查看产生的代码。驱动者从战术上考虑，关注当前编写的代码行。观察者验证语法并从战略上考虑整个程序。他们频繁交换角色，由此产生的代码其编写速度比单个程序员编程要更快且缺陷更少。除此之外，至少有两个开发人员很了解此代码。

什么是结对

让我们看一下典型的代码复审会。一个人员用八小时开发的模块将由八个人员用一小时来复审。最后的结果是 16 个人时花费在一个模块上。但是，复审人员没有足够的时间来熟悉代码，因此他们的复审程度将相当浅。开发人员很熟悉代码，但可能由于太熟悉而无法找到大量缺陷。

将这种做法与结对编程进行比较。如果模块需要两个人员用八个小时来开发，则将花费总共 16 个人时。但是，在这种情况下，将有两个开发人员密切了解代码。一个开发人员看不到的缺陷，另一个可能会看到。

结对编程很简单，但影响是微妙且深远的。结对编程只是编写和复审代码的一种十分有效的方法。如果两个人员很熟悉模块，则代码的缺陷将少得多。代码将具有更好的结构，并且将有两倍的人员了解代码。如果只有这些益处，也已经足够了，但是结对操作还具有更多的好处。

结对人员更胆大心细：单个程序员可能害怕尝试的东西，两个人员将有勇气去尝试并能够对之进行评估。

结对促进团队协作：由于模块不是由一个人员编写，因此代码成为团队（而不是特定开发人员）共用的财产。

结对促进知识的传播：彼此结成对的开发人员越多，在整个团队中传播的系统知识就越多。结果将是团队成员熟悉系统的所有部分，而不是每个成员只了解一个特定部分。

结对提高效率：单独编程的人员经历了兴奋期后，随之而来的将是相对不活跃的时期。结对人员则可以彼此调整步伐。当一个人员感到疲劳时，就交换角色。相比单枪作战，他们可以设法使较高的工作强度保持得更久一些。

结对很有趣：与另一个开发人员一起工作不仅能学到许多东西、具有刺激性，而且很有趣。结对增加了工作满意度并增强了整体士气。

实践

结对

当负责任务的开发人员请求其他人员的帮助时，结对就开始了。规则是当请求您帮助时，必须说是。这不意味着您必须立即停下正在做的工作，而是必须协商可以提供帮助的时间以及作为回报可以获得帮助的时间。

搭档伙伴不承担任务的职责。职责仍由任务所有者承担。搭档伙伴也不承诺与所有者一起工作到任务完成。搭档伙伴只承诺给予帮助。结对中的一个成员变为驱动者，而另一个人员进行观察。驱动者输入代码、运行编译器和单元测试等等。观察者检查每次击键、每条命令以及每个测试结果并提供帮助和建议。双方都一直在忙自己份内的事。

有时，驱动者最了解要做什么，而观察者只是跟着做。而有时，观察者指示驱动者做什么。再有时，驱动者感到受挫时，就将键盘递给观察者，彼此交换角色。还有时，观察者将要求使用键盘，交换角色。在结对中，这些将交替发生。

不断变化的搭档

搭档伙伴不是长期的。一般，结对将持续大约半天。任一成员可基于任何原因而选择离开。当这种情况发生时，任务所有者必须另外寻找一位搭档伙伴。这可能意味着该是任务所有者回报上周与他或她结对的人员的时候了。另一方面，或许他或她应该请求具有相应经验的某个人员来帮助解决特别棘手的问题。

这种不断变化的结对使系统知识能传播到整个开发团队中。在很短的时间内，团队的每个成员都将花时间处理系统的几乎每个部分。这大大减少了项目易手导致的混乱，并使每个程序员在处理整个系统时更加自信。

集体所有权

由于每个人员都将处理系统中所有不同的模块，因此没有人员拥有任何特定的模块。这意味着系统的职责不是按模块划分的，而是整个团队共同负责整个系统。团队中的任何成员可以出于任何原因检出和更改系统中的任何模块。当某一对成员对模块 X 作出更改，而导致模块 Y 的单元测试失败时，这对成员将负责修复模块 Y。

调整步伐和协作

结对编程的交流形式非常激烈。口头对话通常是争论，并且局外人可能很难弄清它的意思。作为局外人，您会听到结队伙伴蹦出比如“分号”或“右括号”这样的词汇。或者在程序员同意或不同意屏幕上显示的内容时，只听到不清晰的咕哝声。两人是如此密切地关注代码，以致于许多沟通可能都是非口头的。身体语言起着重要的作用。搭档知道对方何时对代码不满意（即使他或她什么话也没有说）。一个鬼脸、一声叹气或不安烦躁的情绪 － 所有表现都是合作伙伴之间进行的沟通。有时，一个伙伴握住鼠标，另一个操作键盘。握住鼠标的人员控制即将要行进到的位置，键盘操控者控制在该位置更改或添加的内容。而有时，一个伙伴负责输入，而另一个人则预见即将进行的函数调用并恰在伙伴需要查看说明时在适当的页面打开 API 文档。

当一个伙伴感到疲劳时，另一个可以作为驱动者，让他或她的伙伴通过扮演观察者的角色来得到休息。其他时候，两个伙伴将高度集中精力并频繁交换键盘和鼠标。

概括地说，规则很少并且定式更少。唯一真正的约束是双方必须一直处于忙碌状态并保持紧密的沟通。一个人输入，另一个人望着窗外，这样的一对并不是真正的合作伙伴。

开发人员单独时可以做什么

您不可能在所有时间都处于结对状态。某些项目 － 即采用 *eXtreme Programming*（*XP*）（请参阅参考资料 [1]）流程的那些项目 － 遵循“必须由结对人员编出所有生产代码”这一规则。在那种情况下，当未结对时，您可以检查电子邮件，阅读新的技术或 API 文档，通读不熟悉的代码或者与项目干系人谈论当前的迭代程序或未来的计划。实际上，在未找到结对伙伴的那几个小时，开发人员总是能找到有意义的事情去做。

某些项目对于结对不太严格。有些项目允许单个开发人员编写测试。有些则允许单个开发人员编写抽象类或接口。还有一些只允许开发人员决定结对的最佳时间。不过有一件事情是明确的，不管怎样 － 研究表明当实行结对时，缺陷率会显著下降。

有些人员不喜欢结对

对于有些人员，结对这一概念令他们感到不舒服。根据我们的经验，他们实际尝试结对一周左右后，会发现不适感消失，开始喜欢结对，并发现它很有用。很少有人坚持不喜欢此做法。因此，对于大多数人员，只是尝试它并习惯它的问题。对于真正尝试过并发现仍然不喜欢此做法的那些人员，团队必须找到合适的事情让他们去做。

设备、设施和细节布置

监视器和键盘放置

设备放置对于结对成功是至关重要的。如果伙伴不能坐在彼此的旁边并快速交换键盘，则他们将无法很好地结对。规则是：必须能够互递键盘和鼠标，而不必交换座位。

最佳的安排通常是一张好看的长桌。将监视器放在中间并放置两把面对着监视器的椅子。结对伙伴在监视器两边坐下。请确保两人能很容易地来回移动键盘和鼠标。确保是坐直使用键盘并感到很舒适。还请确保两个伙伴不必旋转监视器就都能看到它。

大房间

要使结对的伙伴易于交换，通常明智的做法是安排大家在大房间工作。在单个房间中布置几对工作站。使用转椅以及漆布或瓷砖地面。将工作站布置成各对都彼此面对。其目的是加强沟通。有时，最重要的沟通是偶然发现的。我们希望增加此类沟通发生的机会。

角落里

目前，许多隔间中工作站都放置在内角处。开发人员面对隔间角落坐着，面前是一个监视器。虽然这对于单独工作很有利，但在这样的环境中很好地结对几乎是不可能的。如果您的环境中有许多隔间，其中工作站又放置在角落里，请在别处安装一些成对工作站 — 或许在会议室里。在会议室桌上使用膝上型计算机进行结对通常是很有效的。

问题和疑虑

结对使工作效率减半

此观点的理由是两个人员一起完成一项任务所耗用的人时数是一个人员完成同一任务耗用的两倍。听上去好像很合理，但实际并不是事实。研究显示（请参阅参考资料 [2]），结对工作损失（如果有损失的话）的生产力极少。研究还显示结对工作充分降低了缺陷率和代码规模，同时大大增加了工作乐趣。

结对伙伴之间的争论

任务所有者在所有设计争论中有最终发言权，但解决争论的最佳方式是尝试两种想法并选择更好的那一个。

专家

传统观念建议研究特定领域（如数据库或 GUI）的开发人员应将他们的精力专门运用在那些领域中。但是在结对编程环境中，这些专家将成为其他并不擅长这个领域的人员的导师。开发人员可以签约所从事专业之外的任务并从专家那里获得帮助。这样，专家的知识可以在项目团队中传播，大大减少了项目易手引起的混乱。

噪音

结对成员在编程时会产生噪音。充满结对成员的大房间将保持不断的、低低的嗡嗡声。有些人员感觉这些噪音令人心烦且容易令人分散注意力。但这未被证实是一个严重的问题。如果您发现噪音令您心烦意乱，可以先到休息室呆一会儿。

不讨人喜欢的家伙

许多团队发现他们“骄傲地”拥有一个或两个不讨人喜欢的程序员。这些程序员做得比其他任何人员都快，无法与其他人员一同工作，并且不允许其他任何人员读他们的代码（或即使允许也读不懂）。

对付这些开发人员的最佳方法是将他们调出项目或换为不编写生产代码的角色。或许他们可以编写短期工具或做一些折磨人的测试工作。

实际障碍和障碍类型

有些人使用标准键盘，而有些则喜欢 DVORAK。有些人需要特殊的键盘、鼠标、显示器、脚踏开关等等。有些人没有耳机和大声的音乐就无法编程。有些人必须用空的 Twinkie 包围绕自己。有些人喜欢用 emacs。有些人喜欢用 VI。还有些人则想用 WordPad 工作。实际上，可以说出的障碍是数不清的。但每个障碍都可经过一点思考和妥协来解决。如果被此类事情阻碍，则这个团队将只是任何事都不会成功的团队。

团队如何计划结对

我们认为任务不应该分配给各结对组，而是每个开发人员应当负责一组任务。我们希望各结对组是随意形成的。每个开发人员继续履行自己的职责，并请求其他开发人员给予暂时的帮助。任务的所有者保留所有权和责任，搭档伙伴只是助手。每个开发人员在估计任务完成时间时，必须考虑与其结对的人员的时间。

结论

结对编程被很好地证明是一种广为接受的代码复审备选方法。而且，它是完全不同的软件编写方法。其好处不仅是大大地提高了工作效率和质量，还影响到了诸如团队实力和士气等等。

参考资料

[1] *eXtreme Programming eXplained*，Kent Beck，Addison Wesley，2000 年。

[2] *Strengthening the Case for Pair Programming*，Laurie Williams，犹他州大学，2000 年 7/8 月，IEEE Software。

Rational
the software development company

两家总部：

Rational Software
18880 Homestead Road
Cupertino, CA 95014
电话：(408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
电话：(781) 676-2400

免费电话：(800) 728-1212
电子邮件：info@rational.com
Web：www.rational.com
全球网址：www.rational.com/worldwide