

TURING 图灵程序设计丛书

PEARSON

The Clean Coder

A Code of Conduct for Professional Programmers

程序员的职业素养

[美] Robert C. Martin 著
章显洲 余晟 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

The Clean Coder

A Code of Conduct for Professional Programmers

程序员的职业素养

[美] Robert C. Martin 著

章显洲 余晟 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

程序员的职业素养 / (美) 马丁 (Martin, R. C.) 著;
章显洲, 余晟译. — 北京: 人民邮电出版社, 2012. 9

(图灵程序设计丛书)

书名原文: The Clean Coder: A Code of Conduct
for Professional Programmers

ISBN 978-7-115-29157-8

I. ①程… II. ①马… ②章… ③余… III. ①程序设
计—工程技术人员—职业技能 IV. ①TP311.1

中国版本图书馆CIP数据核字(2012)第188421号

内 容 提 要

本书是编程大师 Bob 大叔 40 余年编程生涯的心得体会, 讲解成为真正专业的程序员需要什么样的态度、原则, 需要采取什么样的行动。作者以自己以及身边的同事走过的弯路、犯过的错误为例, 意在为后来人引路, 助其职业生涯迈上更高台阶。

本书适合所有程序员, 也可供所有想成为具备职业素养的职场人士参考。

图灵程序设计丛书 程序员的职业素养

◆ 著 [美] Robert C. Martin

译 章显洲 余晟

责任编辑 朱 巍

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京隆昌伟业印刷有限公司印刷

◆ 开本: 700×1000 1/16

印张: 13.5

字数: 207千字

2012年9月第1版

印数: 1-5 000册

2012年9月北京第1次印刷

著作权合同登记号 图字: 01-2012-3060号

ISBN 978-7-115-29157-8

定价: 49.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

Authorized translation from the English language edition, entitled *The Clean Coder: A Code of Conduct for Professional Programmers*, 9780137081073 by Robert C. Martin, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2011 by Pearson Education Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2012.

本书中文简体字版由 Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

题 献

1986年至2000年期间,我与在Teradyne公司时的同事Jim Newkirk成为亲密合作的工作搭档。我们两人都醉心于编程和整洁代码,花了很多时间尝试各种不同的编程风格,把玩各种设计技术。我们还在一起构思各种商业设想。最终,我们一起创办了Object Mentor公司。在和Jim共事的过程中,我从他身上学到了很多东西。其中最重要的,是他对于职业道德的态度,这一点也是我一直努力效仿的。Jim是一名专业人士。我以曾与他共事合作而深感自豪,视他为良师益友。

负阴抱阳，知行合一

“师者，所以传道授业解惑也。” Robert C. Martin，软件开发社区中亲切地称他为 Bob 大叔，正是这样一位明师。

2003 年，他的《敏捷软件开发：原则、模式与实践》(下称 ASD) 在国内上市。我那时进入软件开发行业刚刚一两年，这本书真可谓是及时雨。在精读全书和细心对照书中案例练习后，我感觉自己在面向对象设计方面的功力有了比较明显的提升。那时因工作环境所限，身边没有能够手把手给予技术辅导的导师，因此，那时在我心中 Bob 大叔无疑就是一盏指路明灯。后来在网上找到了不少 Bob 大叔的演讲 PPT，沿着链接，又找到了 Object Mentor 公司其他一些软件开发专家的演讲 PPT 和博客，我如饥似渴地阅读揣摩。现在回头想来，正是在这个阶段我开始建立起“编程技艺”的视角。

时间过得很快，转眼就到了 2010 年，不觉中我已在软件开发的多个领域工作了近 10 年。2010 年，Bob 大叔的《代码整洁之道》一上市，我马上给自己和项目团队订了好几本。在为 ASD 所写的序中，Bob 大叔写道：“最好的软件开发人员都知道一个秘密：美的东西比丑的东西创建起来更廉价，也更快捷。构建、维护一个美的软件系统所花费的时间、金钱都要少于丑的系统。……美的系统是灵活、易于理解的，构建、维护它们就是一种快乐。”如果说 ASD 中更多的是 OO 设计思想和模式精髓的阐述，那么在《代码整洁之道》中，Bob 大叔提供了更为详尽的微距视角，涉及“命名”、“函数”、“代码格式”、“异常处理”、“单元测试”等编码主题，巨细靡遗地向软件工匠们极力传授整洁编码的艺术，进一步向软件开发社区慷慨分享了他在探索“软件之美”旅途中的参证心得。

但是，细心的读者可以发现，在前述两本书中 Bob 大叔阐述的主体还是软件编码技术本身，作为一门技艺而言，止步于具体技术或曰“术”的层面，应该还

未算得完整。后来，在 YouTube、SlideShare 和 Object Mentor 等站点上，我看见 Bob 大叔有不少演讲趋向于聚焦在编程主体即软件开发者自身行为模式和特质层面上，就猜到他不久应该就会有此方面的新著推出。Bob 大叔就是 Bob 大叔，在探索和分享软件技艺的路上，他内心怀有对软件开发社区发展责无旁贷的使命感。果不其然，他将这些体悟浓缩在又一本新著上，这本书便是读者手上的这本《程序员的职业素养》。

本书阐述的是 Bob 大叔关于软件技艺主体的沉思，这些沉思并非是纯粹形而上的思辨推演，而是他对自身编程生涯的深刻反思和经验沉淀。在这本书中，Bob 大叔并非是以高人一等的凌人盛气（事实上，他应该有这样的资格）大行说教，而是毫不掩饰自己在职业生涯中曾犯下的各种错误和不堪往事，以这些案例为载体，现身说法，娓娓道来使自己得以转变和提升的种种“机锋”，并留有意味深长的空间，供读者自己结合自身状况进一步体悟提炼，而非给出硬梆梆的一堆结论。这是何等的胸怀、格局和智慧！

按照传统的太极阴阳思维来看，如果说 ASD 和《代码整洁之道》中的内容是硬性的、技术性的、显性的，故而可以归为“阳”的范畴，那么本书中的“专业主义”、“技艺之道”便是软性的、哲学性的、隐性的，故而可以归为“阴”的范畴。“孤阴不生，独阳不长”，《老子》说“万物负阴而抱阳，充气以为和”。Bob 大叔这三本书为何都选择星云图片作为封面呢？我忽有顿悟。

严肃地选择以软件开发为自身职业方向的软件工程师（我更喜欢称为“软件工匠”）们，如果你同我一样，此前感觉颇为受益于 Bob 大叔的谆谆教诲，那么请不要错过本书。将本书和 ASD、《代码整洁之道》并列案头，三书互为参照，一并静心细读、揣摩体悟、时时对照、检验调整，做到负阴抱阳、知行合一，相信定能渐入佳境。

本书本该早日完成面市，但因我个人方面的一些原因，翻译进度耽搁时日颇多，使我内心极为惴惴不安，这点必须向各位读者，尤其是 Bob 大叔的粉丝和在从事软件开发过程中遇到困顿境地的读者们，深深致歉！

感谢图灵公司的各位编辑对我的信任、宽容和指导。我对图灵公司在技术专

著引入和传播事业上的孜孜以求与坚持不懈的精神十分敬佩！祝愿图灵公司和图灵社区越办越好！

感谢蔡煜兄 (@larrycai) 对几章译稿的前期试读，并提出了不少改善的地方。（我在 2010 年上海 ScrumGathering 会议上结识蔡煜，此后从他的博客和微博上受益不小。）

感谢余晟兄作为审校者为本书付出的辛勤汗水，你的修订和润色使本书的正确性、可读性有了本质性的提升。如果没有你，我不知道自己是否具备勇气将译稿交付出版。在进度滞后的情况下，你又慷慨接受我的邀请成为合译者，在繁忙的工作之余拨冗翻译了本书的第 6、7、9、10 章，使得本书得以早日和中文读者见面。你是我遇见的专业人士的典范，我从你身上学到许多东西。谢谢！事实上，我觉得你才是本书的第一译者！（但人非圣贤，如书中难免存有错讹之处，那是我的责任。）

最后，必须感谢支持我完成艰苦翻译工作的家人，包括我的父母、爱妻 Jenny 和胞妹 Agnes 以及小儿多多。占用了许多本该属于你们的周末时间，我心中深感愧疚，但你们总是很宽容地表示理解，这又使我感觉欠你们更多。

最后，祝读者能够开卷有益，相信作为作者的 Bob 大叔、引进者的图灵公司和译者的我们，定然感到十二分的欣慰。

章显洲

杭州，2012/7/30

享受职业素养

我在招聘时常问的一个问题是：在你过去的工作中，遭遇过哪些印象深刻的困难，最后是怎么解决的？依我的经验，简历写得再漂亮的人，如果这个问题答不好，大都可以直接忽略。为什么会有这种结论？因为我们需要招聘的不是“经历丰富”的人，而是“有职业素养的人”。你遇到的问题可能很容易也可能很难，但我看重的并不是问题的难度，而是解决问题的方式、步骤以及反思的深度。拿恢复误删数据来说，这可能算非常简单的任务。我更感兴趣的是怎样分析问题，找了怎样的资料，采取了怎样的步骤，此后做了哪些措施来避免这种错误再次出现。在我看来，相比问题本身的难度，解决问题的方式和步骤以及反思的深度，都体现出一个人的职业素养。

是的，上面我两次提到了“职业素养”。相比起“专业主义”、“职业化”等说法，我更喜欢用它来翻译 Professionalism，因为素养强调的并不是天赋的神秘，也不是技艺的高深，而是持续积淀的结晶：一方面，它体现了能力和素质；另一方面，它又强调了持续的积累和养成。作为职业开发人员，基本技能不够熟练，当然谈不上职业素养。如果只会迅速地编写代码，却不关心代码背后的意义，不能迅速判断、解决程序运行中的各种问题，不能自信满满地为自己交付的程序承担责任，同样是与职业素养绝缘的——许多所谓的“高手”，其实正是缺乏职业素养的典型。

当然，这只是我对于“职业素养”的理解。由个体经验总结的“职业素养”，多有一鳞半爪的嫌疑，所以即便你觉得上面说的有道理，难免感觉只见树木，不见森林。其实真正的“职业素养”绝不限于上述几方面，而是要广阔得多，深刻得多。要想一窥技术人员“职业素养”的全貌，已经有很多现成的资料可以参考，本书就是其中的佼佼者。

作为一本技术类书籍，本书中有相当的内容是介绍纯技艺的方面，比如测试

驱动开发等，自认已经算“职业开发人员”的人，大概对此并不感冒（不过，我仍然建议你认真看看）。但其他的内容，绝对值得你感冒，比如：什么情况下应该对业务部门说“是”，说“是”意味着什么。如果你没有想过这些问题，或者没有明确的答案，不妨看看 Bob 大叔是怎么说的：

（说“是”时）你对自己将会做某件事做了清晰的事实陈述，而且还明确说明了完成期限。那不是指别人，而是指你自己。你谈的是自己会去做的一项行动，而且，你不是“可能”去做，也不是“可能做到”，而是“会”做到。

就我所见，技术人员经常太容易说“是”，往往在没有明确目标和期限的情况下，就草率给出了确认的答复，而且并不将其视为自己的一种承诺。屡见不鲜的项目延期，有相当原因就是在这种不负责任的情况下说“是”所致。但是我们想想，似乎没有哪一个正经行业，会把不能完成任务的人视为“有职业素养的人”，软件行业也不能例外。

如果你觉得自己已经足够负责，懂得“是”背后所蕴含的意义和责任，也不过如此，我们不妨更进一步，看看关于说“否”。在第 2 章，Bob 大叔介绍了两个项目搞砸的经过。他并没有像常见的所谓专家那样故作聪明地指出实施过程中出现了哪些问题，导致了失败，而是一针见血地指出：这两个项目之所以会搞砸，因为开发人员没有坚决抵制各种不专业的需求（比如一些无关紧要但成本巨大的需求），抵制各种不专业的行为（比如为了赶工期而降低对程序质量的要求），最终只好喝下自己酿出的苦酒。对此，Bob 大叔总结道：

有时候，获取正确决策的唯一途径，便是勇敢无畏地说出“不”字……
我们要明白，委屈专业原则以求全，并非问题的解决之道。舍弃这些原则，只会制造出更多的麻烦……

对我来说，这真是振聋发聩的号角。而且，这种思维，这种视角，其实是许多技术人员所不屑或者不愿面对的——最初我也这么认为，但尝试在工作中主动说了几次“不”之后，我逐渐发现：花三分的力气去抵制无理的需求，可以节省

十分甚至二十分的开发时间；相反，自欺欺人地说服自己凑合接受了无理需求，往往会非常被动乃至无法脱身，到最后，下场就和著名的 IBM OS/360 一样，越挣扎，巨兽在泥潭中就陷得越深。

要学习这样的道理，当然也可以参加培训班，听取授课或者阅读讲义，但那未免太显正经而缺乏亲和力。Bob 大叔的特别之处在于，他总是可以通过浅显易懂的故事，清晰而敏锐地揭示问题的核心所在。其中许多故事正是他自己亲身经历的，阅读过程中常会会心一笑，因为遇到了开发人员都懂的妙趣，比如费尽全力也是徒劳，无法让其他人理解“编辑程序的程序”。笑过之后，又会认识到许多道理——无法让其他人理解“编辑程序的程序”并不是真正的原因，真正的原因是：“客户……对功能的设想，其实经不起电脑前真刀真枪的考验……问题在于，东西画在纸上与真正做出来，是不一样的。业务方看到真正的运行情况时就会意识到，自己想要的根本不是这样。一看到已经满足的需求，关于到底要什么，他们就会冒出更好的想法——通常并不是他们当时看到的样子……真正的解决办法，是约定共同认可的验收测试标准，并在开发过程中保持沟通。”至少就我的经验，这一点是说得非常对的。我曾经尝试在与业务部门确定目标原型之后，要求对方指派一个负责人在 IT 部坐班，负责协商、跟进整个开发流程，确认每一点修改。这样既保证最终结果符合业务部门的需求，又提高了开发人员的工作效率，综合来看，成效非常显著。

类似的例子还有很多，在阅读这本书时，我经常惋惜：如果早一点读到这本书，或许我之前就不会犯这样那样的错误，就能更早更好地积累自己的职业素养。况且能有妙趣横生的书讲述看似枯燥的“职业素养”，对读者来说，又是一种幸运。德国作家托玛斯·曼曾经津津乐道于“斜躺在沙发上整天阅读叔本华”的美妙感觉，那是因为叔本华的文笔优美、流畅，可以把哲学变为惬意的享受。作为同时读过叔本华和 Bob 大叔的人，我想说，斜躺在沙发上整天阅读《程序员的职业素养》，认识和了解开发人员的职业素养，同样是相当惬意的享受。

余晟

广东，2012/7/18

序

你选了这本书，那么我不妨认为你是一名软件工程师。很好，我也是。既然如此，我得和你谈谈，我为什么会读这本书。

事情是这样的：不久以前，在不远的地方。来，背景、灯光、摄像、演员，一切就绪，那我就开讲了……

几年前，我在一家中等规模的公司工作，公司销售的是政府严格管制的产品。这类公司你肯定知道：办公场所位于一幢三层小楼里，我们坐的是格子间，总监级以上人物则可以拥有私人办公室；把相关人员召集到会议室里开个会，可能得花一个星期左右的时间。

我们所在的市场竞争非常激烈。这时，政府放开了一个新产品。

突然间我们拥有了一批全新的潜在客户，我们所要做的就是让他们购买我们的产品。这意味着必须在某个截止日期前向联邦政府提出申请，在另一个截止日期前通过评估审计，并在第三个截止日期前完成交付。

管理层一遍又一遍地向我们强调这些截止日期的重要性。如果延期一次，政府就会把我们列进当年的黑名单；而如果客户们没有在第一时间和我们签约，他们全部就会和别的供应商签。如果拿不到订单，我们就彻底出局了。

在这种环境下，有些人会抱怨，有些人则会说：“艰难困苦，玉汝于成。”

我那时是从开发部门晋升上来负责技术方面工作的项目经理。我的职责是确保网站按期上线，这样潜在客户便可以从网站上下载资料，其中最重要的资料是注册表单。负责业务方面的项目经理是我的搭档，我叫他 Joe。Joe 的工作职责是销售、做市场推广和分析非技术性需求。这位老兄平素也挺喜欢用“艰难困苦，玉汝于成”这样的调调说事。

如果你在美国公司工作过，肯定知道指责、过失追究和工作厌恶症都是司空

惯的。不过在我们公司里，Joe 和我有个十分有效的办法来解决这类问题。

我们这对搭档有点儿像蝙蝠侠与罗宾，任务就是要把事情搞定。我每天都会和技术团队会面；我们每一天都会调整计划，找到关键路径，扫除在关键路径上所有可能出现的障碍。如果有人需要什么软件，我们就去拿来。如果有人一边抱怨“天哪，我该吃午饭了”一边说自己更“喜欢”配置防火墙，我们就会给他们带午餐。如果有人想去搞定配置问题，但手上还有其他更重要的事项要去处理，Joe 和我便会去找他的主管协调。

如果不行，就去找经理。

再不行，就去找总监。

总之，肯定会把事情搞定。

如果说我们常常大动肝火、踢翻椅子、以大吼大叫的方式来沟通，这可能有点儿夸张了，但我们确实使尽了浑身解数来搞定遇到的每件事情，并且发明了一些新方法。但是，至今为止，让我引以自豪的是，我们并没有违背职业道德，没有为达目的不择手段。

我认为自己是团队的一员，而非凌驾于团队之上、只会中途插进去写一句 SQL 语句或做点儿结对编程写一两句代码的那种人。当时，我也是如此看待 Joe 的，认为他也是团队成员之一，而不是凌架于团队上，置身事外。

后来我才明白，Joe 自己并不是这么看的。对我来说，这是令人很郁闷的一天。

那是个星期五下午的一点钟，下周一一早，网站就要按计划上线了。

完工了。*DONE*。每个系统都已就绪，我们已经准备好了。我把整个技术团队召集在一起，召开最后一次 Scrum 会议，准备上线。与会的不只是技术团队，还有市场营销、产品负责人这些业务人员。

我们感到十分自豪。这真是个美妙的时刻。

这时，Joe 进来了。

他大致是这样说的：“有些坏消息。法务还没准备好注册表单，因此网站还不能启用。”

这没什么大不了的。在整个项目过程中，我们时常受阻，不是因为这样的事

就是因为那样的事，但是，对付这些障碍，我们的“蝙蝠侠/罗宾”这招却屡试不爽。我对此已有对策：“好的，伙计，我们再试下老方法。法务在三楼，对吧？”气氛不太对头。

Joe 并没有同意我的提议，而是反问道：“Matt，你是什么意思？”

我说：“你知道的。我们的经典配合。我们现在谈论的就只是四个 PDF 文件，对不对？这些 PDF 文件已经准备好，就差法务批准下就可以了，是不是？我们只要在他们旁边盯牢他们，应该就可以把这件事情搞定的！”

Joe 不同意，他回答说：“我们下周只需迟些时候上线网站就可以。没什么大不了的。”

你可能已经猜到了后续的交谈，大致像下面这样。

Matt：“但是为什么呢？他们只需要几个小时就可以搞定的。”

Joe：“几个小时可能不够。”

Matt：“但是他们整个周末都可以处理啊。时间很充裕。就这么办吧！”

Joe：“Matt，这些法务都是专业人士。我们不能逼迫他们，没理由要求他们为了我们这个小小的项目而牺牲个人时间。”

Matt：（暂停）“……Joe……那你对过去这四个月来我们对技术团队所做的事情又做何感想呢？”

Joe：“没错，但是这些法务可是专业人士。”

冷场。

深呼吸。

Joe 刚才说什么来着？

那时，我认为把“专业人士”这个词用在技术人员身上最贴切不过。

但现在仔细回想一下，我对此已经不甚确定了。

让我们换一个视角，再来看看“蝙蝠侠与罗宾”技术。我认为我是在鼓励团队努力呈现他们的最佳状态，但我怀疑 Joe 是在和我们玩博弈，他暗地里认为技术人员是站在他的对立面的。想想看：不然又有何必要到处巡视，在依赖别人做

事的同时，又常常闹出踢翻椅子大动干戈这样的事呢？

为什么我们不能去询问团队成员项目什么时候可以完工，在获得确切回答之后就相信这个回答，并且也确实不会因为信任却最终反受其累呢？

当然可以，作为专业人士，我们确实应该做到这一点……但同时，我们确实还做不到。Joe并不信任我们，只有对技术团队进行微观管理才能让他安心。而与此同时，出于某种原因，他确实能够信任法务团队，也并没有要对他们进行微观管理的想法。

这一切说明了什么问题？

法务团队肯定以某种方式展现了他们的专业精神，而技术团队尚未做到这点。

这些团队肯定以某种方式让 Joe 信服，他们并不需要一个保姆一直跟在身边，他们彼此间不是在玩博弈游戏，他们应该被视为值得敬重的合作伙伴。

不，我不认为这和那些挂在墙上的花哨证书，或是他们在学校里多待了些年头有什么关系。尽管这些年来，学校里可能已经开了不少社交培训课程，里面涉及不少与如何表现得更专业相关的内容。

自从多年前的那一天之后，我一直在想，技术人员需要如何改变才能被视为专业人士呢？

哦，我已经有了了一些体会。我已经用博客记录了一些想法，也看了很多相关的内容，我设法以此提高自己的工作生活状况，同时也能帮助其他人有所提升，但我尚未听闻有人在这方面有什么写作计划，把如何成为软件专业人士的全部秘诀和盘托出。

直到后来，有一天，我意外地获得审阅一本书的初稿的机会；这本书，正是你现在手中拿着的这本。

这本书将会由浅入深、详细讲解该如何展现自己，如何以专业人士的方式与人交流协作。没有陈词滥调，也非寻章摘句纸上谈兵之作，里面阐述的都是具体的行动方法和要诀。

其中的某些案例中，作者可谓用心良苦，字字净言。

一些例子中有对话录和总结，甚至还针对别人“无视你”的情形专门提供了

相关的行动建议。

嘿，看，Joe 又来了，我们回到了早前的一幕。

Joe 和我，我们又回到 BigCo，把那个网站大项目重新做一次。

只不过这次，请想象一下，这次的做法有一点点不同。

这次，技术人员不再找借口拖延，而是勇担重任；不再推卸估算工作或置身事外让其他人来做计划（然后对计划抱怨不休），而是真正做到了自组织，并做出了郑重承诺。

现在想象一下，大家能够真正紧密协作起来了。当程序员因为运维方面的问题受阻时，他们会打电话给系统管理员，之后，系统管理员就会马上着手清除障碍。

Joe 也不必大动干戈才能推进解决 14321 号问题；他可以看到 DBA 正在勤奋工作，而不是在网上冲浪。同样，他从技术人员那里拿到的估算结果看起来非常一致，他不会感觉项目在技术人员那里的优先级是无足轻重的。以往所用的试图操控进度的所有招数手段，现在都派不上用场了，技术人员现在不会说“我们尽力而为吧”，而会代之以“这是我们的承诺；如果你想调整目标，请随时联系我们”。

过了一段时间，我想 Joe 就会认同技术团队同样也是专业人士。是的，确实如此。

那么，想要从技术人员晋升为专业人士，该经历哪些步骤呢？本书将为你和盘托出。

祝你迈上职业生涯更高的一个台阶。我想你肯定会喜欢这本书的。

Matthew Heusser
软件过程博物学家

封面图片

封面所用的这张惊艳图片，很容易让人联想到魔王索伦的眼睛，它是 M1 蟹状星云。M1 位于金牛座，在金牛左角尖的天关星右边约一度位置。整个蟹状星云是一次超新星爆发在宇宙中所散开的残骸，它的亮度不逊于木星。在 6500 光年外的地球，古代中国的天文学家观测到了这场爆炸，日期刚好是公元 1054 年 7 月 4 日（美国国庆日）。确实如此，当时人们在白天用肉眼就看得见！在随后的六个月里，它才慢慢地从肉眼能见的视野中淡出。

这张封面图片是由可见光线和 X 射线合成的。可见光图像由哈勃望远镜摄得，作为外层；内中看起来像蓝色靶心的图像是由钱德拉 X 射线望远镜摄得。

可见光图像显示的是超新星爆发时混合着重物质残骸迅速膨胀的尘埃和气体云。此星云现在的直径为 11 光年，质量是太阳的 4.5 倍，并且仍以每秒 1500 千米的速度剧烈膨胀。且不论别的，那次久远年代时的爆发中所产生的动能真可谓强劲。

靶心正中的那个蓝色亮点正是脉冲星所在。正是脉冲星的形成导致恒星爆炸。在那颗濒临死亡的恒星内核，约有一颗与太阳（直径约为 140 万公里）差不多质量的物质聚爆成直径约 30 公里的中子星。聚爆产生的巨大动能，加上中子形成时微中子的猛烈冲击，将恒星撕裂，以爆发宣告了死亡的到来。

脉冲星仍在以每秒 30 次的速度旋转，并且一边旋转一边发光，通过望远镜可以看到闪光。正是由于这种光线脉冲，人们才把它命名为“脉冲星”。