

14 | 项目管理工具：一切管理问题，都应思考能否通过工具解决

2019-03-28 宝玉 来自北京

《软件工程之美》



你好，我是宝玉，我今天想与你分享的主题是：一切管理问题，都应思考能否通过工具解决。

早些年我在做项目管理工作的時候，除了制订计划外，还要花不少时间去跟踪计划的执行情况。

项目管理上出了问题，管理者总是喜欢从流程规范的角度去想办法，于是为此设定了不少流程规范，例如每天要写日报，根据日报更新项目进度，每周要开周例会，看看项目有没有执行上的问题。

对任务进度的量化也是个很困扰项目经理的事情，需要频繁地去问程序员：“你这个任务进展如何，大概完成比例多少？”，从程序员那得到的答复通常都是个很乐观的数字，例如80%。第二天以为他能做完，结果一问是90%，就这样要持续好多天才真的算做完。

所以后来我得出一个结论：**一个任务，只有 0% 和 100% 两种状态是准确的，中间状态都是不靠谱的。**

除此之外，还有个问题就是，项目的进展并不太直观，除了项目经理每天看计划表，对计划有一个大概了解以外，其他人可能只有在到了计划设置的“里程碑”时，才对进度有比较直观的感觉。

项目成员手头事情做完，如果和计划有出入，也不知道自己接下来该干嘛，都要跑去问项目经理，所以项目经理对于很多事情都要从中协调，日常有很多繁重的任务管理工作。

后来我发现其实很多管理者都有类似的困惑：任务不好量化难以估算，项目成员对当前项目进度缺少直观感受，管理者要花大量时间在任务管理上。

这些年，随着软件项目管理工具的发展进化，发现当年困扰我的这些问题已经不再是一个主要问题，因为通过工具就能很好的解决这些问题。

这也是我这些年项目管理和技术管理的一点感悟：

一切管理问题，都应思考能否通过工具或技术解决，如果当前工具或技术无法解决，暂时由流程规范代替，同时不停止寻找工具和技术。

下面的微博即是一例，当遇到问题时，不仅从流程上思考有没有问题，更要考虑是不是可以用工具或技术手段来解决。



宝玉xp V

3月9日 07:40 来自 Mac客户端

👍//@蛙蛙王子:感觉梳理好简洁高效的流程规范，能自动化的都自动化，是技术leader首要任务//@宝玉xp:👍//@郑昀:摒弃单纯依靠员工之间互相提醒、依靠个人认真细致来规避相同错误的固有思路，铁打营盘流水兵，靠人终归是靠不住的，最好靠遵循规则的机器。

@宝玉xp V

刚有人跟我抱怨新人发布一个包时没遵守流程，结果出错了，让我好好教教他们。我看了一下，问题不像他说的那么简单：首先文档没写清楚，漏了一些关键步骤，尤其是验证步骤。另外有些步骤完全不需要全手动，应该花点时间把流程自动化，这样以后就不担心这个问题了。不然就算教好一个，下一个还一样出问题

3月9日 00:12 来自 微博 weibo.com

阅读 1.6万 | 22 | 3 | 3

在这里，我还是先带你看一下项目管理工具软件发展史，通过工具的演化，你可以更深入的了解到工具是怎么解决这些管理问题的。

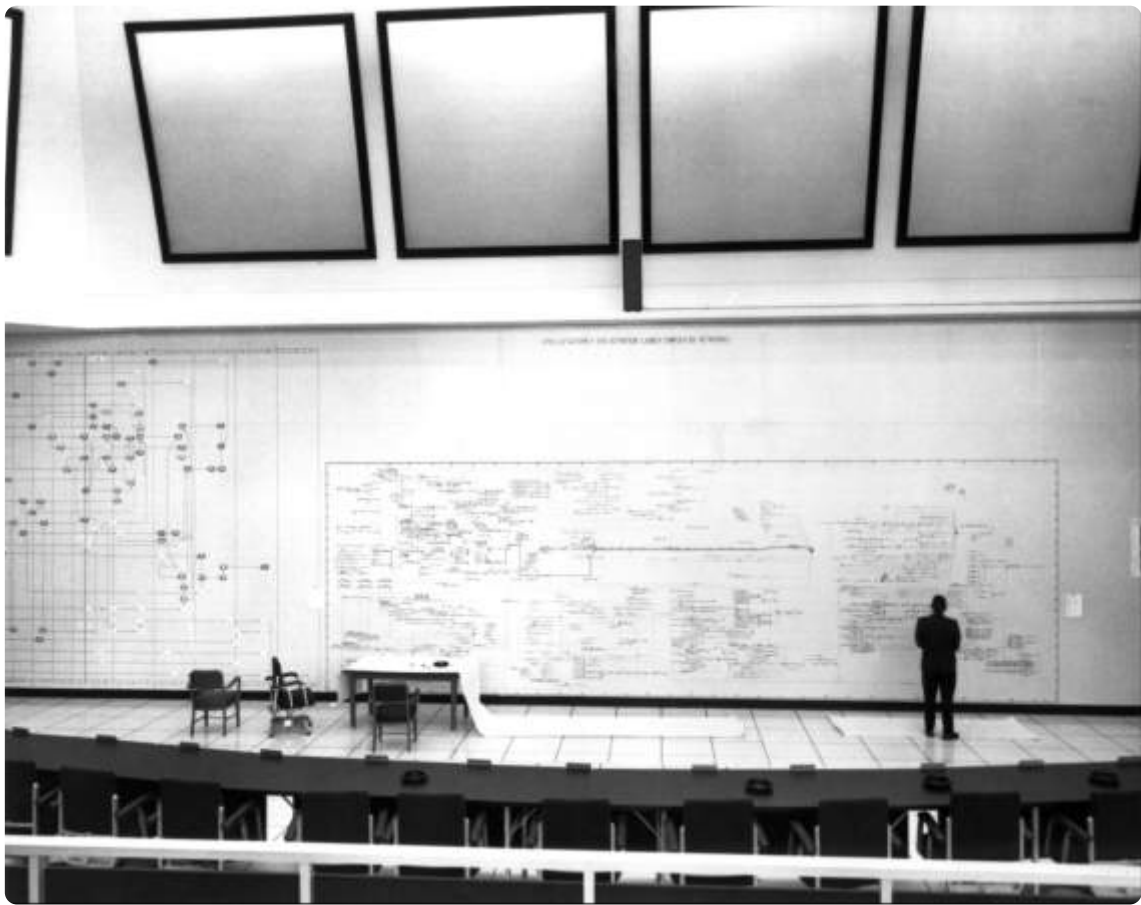
项目管理工具软件发展史

在没有项目管理工具的年代，都是怎么管理项目的？

早些年，我除了好奇过大厂是怎么开发大型软件项目以外，还好奇过像登月这种超大型项目是如何做项目管理的。正好前不久看了余晟老师写的一篇文章《🔗“阿波罗”登月中的工程管理一瞥》，让我有机会一窥究竟。

其实这种大项目的项目管理并不神秘，就是像我们专栏《🔗11 | 项目计划：代码未动，计划先行》那一篇讲的，这种大项目也是采用 WBS（工作分解结构）把所有任务一级级分解，再排成计划，按照计划有序进行。

但阿波罗项目是个超大型项目，所有的任务分成了 A、B、C 三级，到 C 级已经有超过 4 万个任务。要给这四万多任务排出项目计划就太不容易了，一共要几十名分析人员来协调和跟踪所有的任务。最终列计划的图表贴在墙上超过 100 平米。

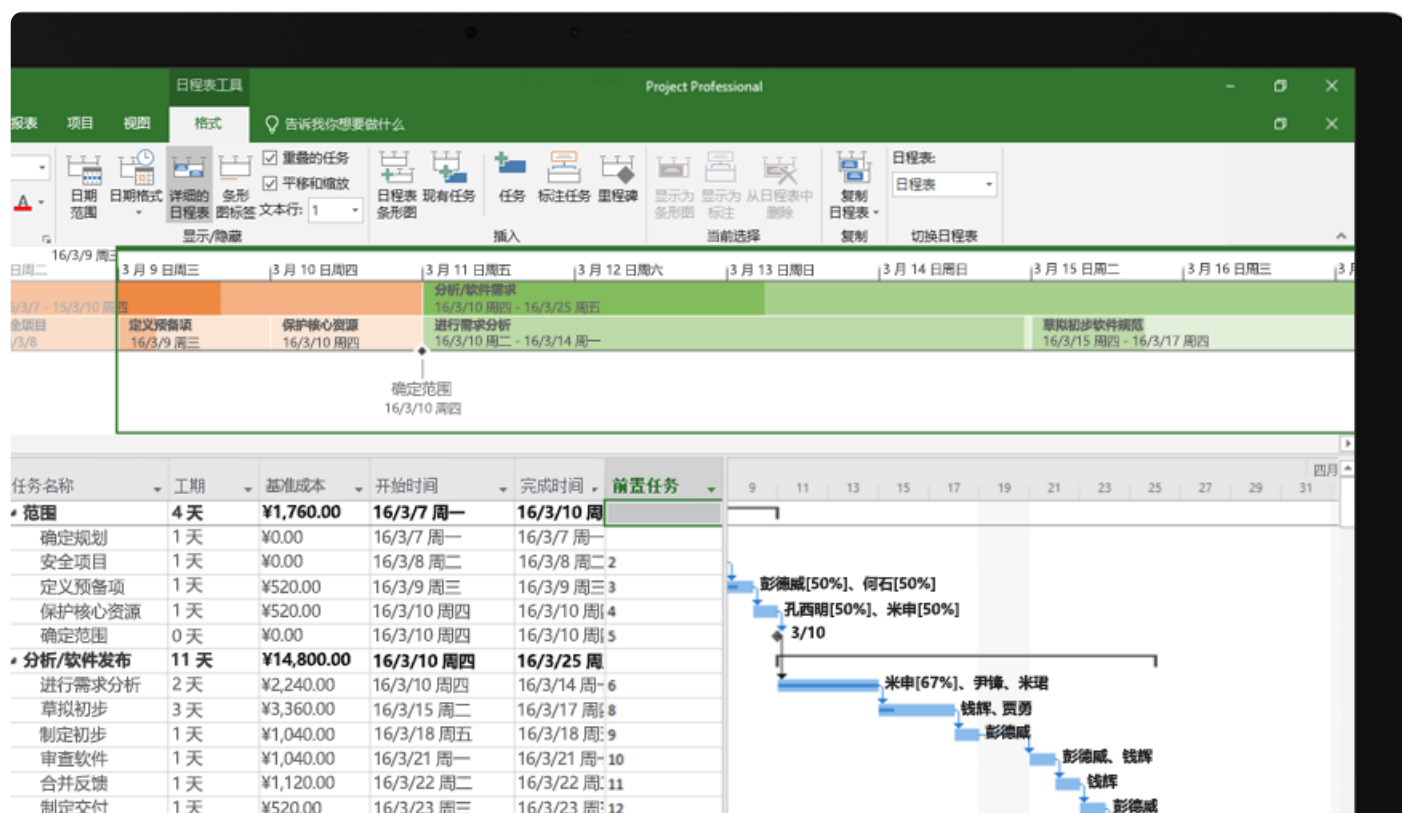


阿波罗登月项目巨型计划图

在没有项目管理工具的年代，要制订一个项目计划非常之不容易，需要专业人士花大量时间，而且每次修改调整，都要再花费大量时间精力。

最初的项目管理软件：项目计划工具

直到后来像微软的 MS Project 这样的项目计划工具软件普及，才让制订计划变成了一个相对容易的事情，可以方便的对分解好的任务排出计划。



图片来源：MS Project官网

早些年软件项目的开发以瀑布模型为主，瀑布模型的这种按阶段划分的开发模式，和 WBS（工作分解结构）这种将任务层层分解的理念不谋而合，MS Project 这种软件可以非常好的将所有任务分解、制订计划，按照计划跟踪执行。所以那时候，会使用 MS Project 就是项目经理的标配。

MS Project 虽然解决了计划制订的问题，但还是有些不足之处。例如不方便跟踪任务进度，进度不直观等。

再加上后来敏捷开发开始兴起，很多项目都开始采用 Scrum 的方式来进行项目管理，开发变成了迭代的方式，以前单纯的项目计划工具，就不能很好的满足项目管理需要了。

基于 Ticket 的任务跟踪系统

传统的项目计划软件还有很多问题无法解决。比如，很多人都有过以下类似的项目经历：

产品经理口头让开发对产品做一点小改动，开发也答应了，后来就把这事忘了，或者测试都不知道还有这事，也不记得要测试这个模块；

代码审查的时候，发现组内某个同事的代码没有写单元测试，但是因为任务紧，只能先上线，于是叮嘱他后面一定要把单元测试代码补上，结果还是忘了。

日常项目中像这样的小事情不少，如果不记下来很容易忘记，如果用传统的项目计划软件排进去又很麻烦，直到后面有了基于 Ticket 的任务跟踪系统，才很好的解决了这个问题。

Ticket 跟踪最早源于客服的工单（Ticket）系统，每次客户接到一个问题，就创建一个工单，后续和客户的每一次交流和处理，都要更新工单内容和状态，直到结束。

最早在软件项目中，应用 Ticket 跟踪系统的领域是测试领域，用来追踪 Bug，后来逐步衍生到整个项目管理领域，不仅跟踪 Bug，还用来跟踪需求、开发任务等。

也有很多系统用 Issue 来表示 Ticket 的概念，无论 Ticket 还是 Issue，表示的都是一个工作任务，可以包括软件的 Bug、功能需求、某个模块的开发、系统的重构任务等。

那一个 Ticket 应该包含哪些主要信息呢？

一个 Ticket，应该包含：

标题：摘要性的描述 Ticket 内容；

类型：属于什么类型的 Ticket：Bug、需求、任务；

内容：Ticket 的详细内容，例如，如果是 Bug 的话，除了要写清楚 Bug 内容，还需要重现步骤。如果是需求的话，要有需求的描述，可能还需要额外的文档链接辅助说明；

创建人：谁创建的这条 Ticket；

优先级：这个 Ticket 的优先级高还是低；

状态：Ticket 的状态，例如：未开始、处理中、已解决、重新打开、关闭等；

指派给谁：这个 Ticket 被指派给谁了，谁来负责；

历史记录：整个 Ticket 改变的历史信息，用以跟踪；

当然除了这些外，还有一些其他信息，例如创建时间、附件、标签、版本等。另外现在的 Ticket 跟踪软件都有强大的定制功能，可以增加额外的辅助信息，例如你是基于敏捷开发，还可以加上 Sprint、故事分数等信息。

Ticket 的这些内容，基本上可以包含一个工作任务所需要的所有内容。有了 Ticket 之后，无论大到一个功能需求，还是小到一个 Bug，从它创建，一直到完成，整个过程都可以方便的被跟踪起来了。再也不担心像任务被忘记等前面提到的这些情况了。

基于 Ticket 去跟踪任务，不再需要通过日报、一对一会议的方式来收集任务执行情况，负责 Ticket 的项目成员在完成任务后，会直接修改 Ticket 的状态，这样其他人就可以看到 Ticket 是否已经完成。

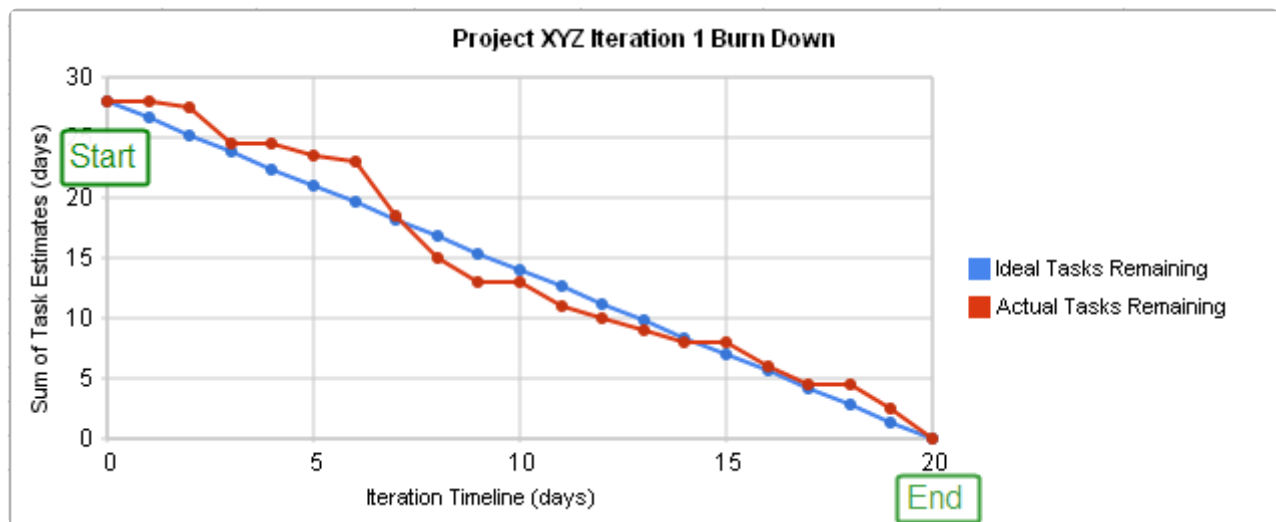
Ticket 通过各种不同状态，例如未开始、开发中、完成等，可以很直观的了解任务的进展，这就避免了任务难以量化的问题。

Ticket 跟踪系统和敏捷开发也是很好的搭档。在敏捷开发中，产品 Backlog（产品待办任务列表）是一个用来放所有产品的待办任务的清单，在每个 Sprint 开始前的迭代计划会议上，从产品待办任务清单里面选取一部分任务到 Sprint 的待办任务清单（Sprint Backlog）中。

当使用 Ticket 跟踪系统后，就可以把所有产品的待办任务用 Ticket 都记录起来，当我们在迭代计划会议上选取好任务后，就标记为要在当前 Sprint 完成，这样后面就可以方便的筛选出属于当前 Sprint 的所有 Ticket，这样大家就可以从 Ticket 跟踪系统知道我们这个 Sprint 有哪些 Ticket 需要完成、进展如何。

如果将当前 Sprint 中，从开始到结束，每天记录一下 Sprint Backlog 中未完成 Ticket 的数量，绘制成一张图表，横轴表示时间，纵轴表示剩余 Ticket 数量，就可以通过图表直观地看到还剩下多少工作。

这种用于表示剩余工作量的工作图表也叫燃尽图（burn down chart），可以直观的预测工作将在何时全部完成。



图片来源：维基百科

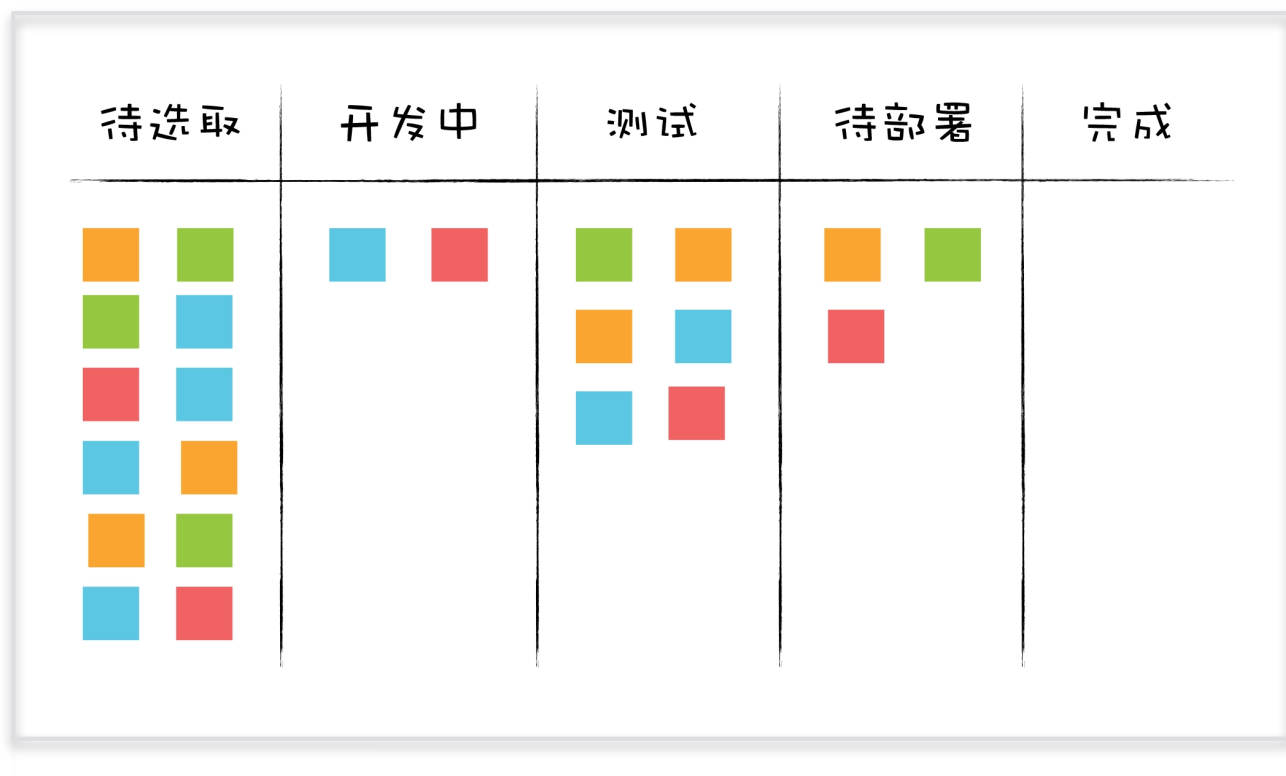
基于 Ticket 的任务跟踪系统，很好的弥补了项目计划工具的不足，让项目中大大小小的各种开发任务都可以方便的记录跟踪起来。燃尽图也可以直观的了解剩余工作情况。

如果说美中不足的话，就是整体的 Ticket 状态还不是很直观，例如不能清楚的看到哪些任务在进行中，哪些任务待领取。

基于看板的可视化任务管理

看板本来是在 1940 年由“丰田汽车”发明的生产管理系统，其中一些理念被借鉴到软件开发中，尤其是其可视化的任务管理方式，很好地解决了早期 Ticket 跟踪系统不直观的问题。

所以现在的 Ticket 任务跟踪系统几乎都会有看板视图，通过看板可以很直观的看到当前任务进展情况。



参考上图，可以看出，在看板视图上的所有 Ticket，可以很直观的看出哪些还没开始，哪些进行中，哪些已经完成。

这种可视化的任务视图，不仅是对项目经理，可以很直观看到进展，对于普通项目成员也是很方便。

从“待选取”栏选择一个 Ticket，拖动到“开发中”栏，表示这个 Ticket 已经选取，开始开发了。

手头上的 Ticket 开发完成后，就可以将 Ticket 拖动到下一栏——“测试”栏。

测试人员看到新加入“测试”栏就可以从测试栏选取 Ticket 进行测试。

如果测试没通过，Ticket 就会被拖动到“待选取”栏。

如果测试通过，Ticket 就会被拖动到下一栏——“待部署”栏。

部署完成后，所有“待部署”栏的 Ticket 就会被拖动到“完成”栏。

整个过程完全不需要项目经理从中协调太多，尤其是结合每日站立会议，可以让项目成员自发有序地按照看板开展日常工作。

借助 Ticket 跟踪和看板可视化，项目经理可以从繁重的任务管理中解放出来，可以抽出来时间做一些其他更重要的事情。

以上就是项目管理工具的一个演化简史，可以看到，每一次工具的发展进化，相应的很多项目管理工作就可以得到简化，很多早期的项目管理问题，也就不再是问题了。

有哪些项目管理软件可以选择的？

在了解完项目管理工具的发展历史后，再给你介绍一些目前国内国外主流的项目管理软件，帮助你根据自己项目需要进行选择。

如果单纯是项目计划工具，功能最好、最全的应该是微软的 [🔗 MS Project](#)，但遗憾的是只能运行在 Window 上，不支持 Mac 平台。如果要在 Mac 上使用项目计划工具，可选的有 [🔗 OmniPlan](#)和 [🔗 Merlin Project](#)。

而且这些项目计划工具，现在也都支持了看板视图。不过如果只是单机支持的话，意义并没有那么大，需要在线版的 Ticket 跟踪结合看板视图，才能让整个团队可以一起浏览操作，发挥其最大效用。

基于 Ticket 的任务跟踪系统，最有名的应该是 [🔗 Atlassian](#)公司出品的 [🔗 Jira](#)软件，功能全面，体验很好。Jira 主要是在海外比较流行，因为访问速度和使用习惯等原因，国内用户要相对少一些。

同类产品也很多，微软的 [🔗 Azure DevOps](#)（以前叫 TFS, Team Foundation Server），和微软系的产品如 Visual Studio、Azure 可以很好的整合。

代码托管平台 [🔗 GitHub](#)本身也集成了一套 Issue 跟踪管理系统，虽然没有 Jira 那么强大，但是对于普通项目来说，足够用了。尤其是对于开源项目，完全可以基于 GitHub 的 Issue 进行日常的项目管理。

国内同类的软件有：

🔗 禅道：为数不多提供开源版本可以自己搭建的；

🔗 Worktile：集成了即时消息软件；

🔗 TAPD：腾讯出品，可以和腾讯的服务很好整合，例如企业微信和腾讯云；

🔗 云效：阿里巴巴出品，可以和阿里的服务很好整合，例如阿里云和钉钉；

🔗 DevCloud：华为出品，和华为云有很好的整合。

还有一些其他产品，这里就不一一列举。

那么该如何选择适合的工具呢？

从功能上来说，基本上，上面提到的每一款产品都能满足日常项目管理的基本需求，建议从项目特色、团队成员、价格和服务等因素综合考虑。

例如说你的项目完全是微软技术栈，就可以考虑使用 TFS；如果你深度使用阿里云和钉钉，那么就可以考虑阿里的云效；如果你想自己搭建，那么就可以考虑 Jira 或者禅道。

这些产品都有免费版本，可以先试用，你可以仔细对比后，根据自身的情况再最终决定。

总结

今天我带你一起了解了软件项目管理工具的发展历史：从完全手工方式管理项目，到借助计划工具分解安排计划，到基于 Ticket 跟踪管理任务，再到基于看板的任务可视化。每一次工具的升级，都是对项目管理工作的一次简化。

合理的使用项目管理工具，可以帮你极大提高管理效率，起到事半功倍的效果。我也列举了一些目前国内外主流的项目管理工具，希望可以帮助你做出选择。

最后，对于日常项目管理的问题，你也可以多思考是不是可以由工具或者技术手段来解决的。

课后思考

你在日常项目中，有哪些应用工具或者技术解决项目问题的例子？或者你觉得可以用工具或技术解决的问题？你现在项目中用的是什么项目管理工具，有什么优缺点？欢迎在留言区与我分享讨论。

感谢阅读，如果你觉得这篇文章对你有一些启发，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (32)



J.M.Liu

2019-03-28

我觉得辅助计划工具是从项目规划和任务分解出发，以任务之间内在逻辑关系为依据组织任务，优点是能够清晰地看到整个项目的蓝图，缺点是结构化程度太高，不够灵活，不能适应项目执行期间遇到的变化。基于ticket的管理跟踪系统是从项目执行的角度出发，以执行周期为依据组织任务(如一个sprint)，注重任务的状态跟踪，优点是灵活，缺点是缺乏结构化，各任务之间的关系不明确，容易只见树木不见森林，因此不适合做项目规划和任务分解。因此，需要将二者结合起来用，在规划和任务分解阶段，用项目规划工具，生成蓝图，最后把分解后的任务做成一个个ticket，做项目跟踪

作者回复：给你点赞👍

你说的这一段是我文章总结所欠缺的部分，Ticket跟踪系统还不能完全替代项目计划工具，需要结合使用。



👍 40



易林林

2019-03-28

完全手工方式管理的优点在于自由空间大、项目结构松散，比如临时添加需求、临时添加人员、临时改变策略等。一旦管理者没有足够的能力去驾驭项目的整体架构，随着项目时间的推移，项目不是越做越简单，而是越做越难，可能到处都是窟窿，根本没法持续下去，并且责任和义务大部分集中于项目管理者。

尽量采用软件工具管理的优点在于对需求、人员、进度、里程碑等可以进行事无巨细的分解或者组合，明确每个人的职责，明确每件事完成的要求，既可以让参与人员看到长期目标，也可

以让他们看到短期目标，而不是遥遥无期。可以这样讲，没有路标的100公里总是比有路标的100公里来得费尽得多，还有就是很容易让参与者失去信心，丧失斗志。

宝玉老师在上面提到了部分工具，能否把项目管理每个阶段用到的典型工具分享一下，谢谢。

作者回复：总结的非常好👍

其实每个阶段都有关于工具的章节，比如这一篇就是项目规划篇的项目管理工具。

需求分析篇的工具要讲原型设计，需求阶段还有需求收集管理工具，通常可以用Ticket管理系统（如Jira）、源代码管理（如git）或文档管理工具（如Google Docs/石墨文档）来做。

设计阶段其实主要用文档工具，用MS Visio/PPT画图。

编码阶段主要是源代码管理工具、各种IDE、持续集成平台（Jenkins）的搭建

测试阶段主要是有测试用例管理系统（例如TestRail），有Bug跟踪系统（基本上和项目管理工具一起的，例如Jira）

运维监控有日志管理系统（例如ELK），监控（例如Wavefront），报警（例如PagerDuty）



👍 17



熊斌

2019-10-31

之前我们项目经理是从IBM出来的，非常擅长使用Excel,我们的项目wbs都是他用VBA做的工具。

不足之处是，无法有效追踪项目进度。

追踪进度的时候，需要问参与的相关人员完成情况。作为开发，我要是完成了20%，为了数据好看一点，我可能会报50%.....

作者回复：进度跟踪时，拍脑袋想一个进度百分比这种我也经历过，开始百分比进展很快，然后80%之后就越来越慢了，90%可能都好多天才能100%。

所以说：一个任务，只有 0% 和 100% 两种状态是准确的，中间状态都是不靠谱的。像看板这种只有“TODO”、“进行中”、“完成”等这样几种状态还是更科学可行。



👍 5



Charles

2019-03-29

我们在用云效，用云效之前用过tower主要用看板和任务管理，还自己搭过jira之类的

云效相对来说更加健全一些，主要解决需求管理、版本任务、bug、测试用例、代码管理、持续部署等大部分项目管理的问题

优点：因为用他云服务，所以整合度挺好的

缺点：因为不算他特别核心业务，所以感觉细节问题还挺多的，部署也经常失败，但是最近好像有改善

另外，问老师一个问题，文章中讲到ticket做完到待测试，这一步测试人员是否马上应该跟进测试，但是很多ticket其实是有关联的，所以到底是一个版本计划内的任务全部完成再测试还是一个一个ticket分开测试？如果是单个ticket测试的话，这个ticket是否需要保证和其他模块无关联或关联性较少？

作者回复: 文中只是一个示例，可以针对性调整，比如你可以增加一栏：开发完成。对于完成但不具备测试条件的先放到开发完成栏，等到相关ticket都开发完成，再一起放到待测试，这样会更准确。



👍 5



胖虫子

2019-04-22

说的好好，但在现实中，往往只有最后一个完成时间，明明完不成，硬上，项目经理就是天天催

作者回复: 实际项目中确实有很多残酷的现实，而我们学习软件工程，不就是为了知道理想的开发软件是什么样子，好的开发方式是什么样子，然后超那个方向努力么！



👍 4



纯洁的憎恶

2019-03-29

工具是流程的进阶，它使流程规范真正发挥作用的同时，将其“副作用”控制在合理范围内。

Ticket、可视化看板等工具灵活、便捷、高效，不仅可以用于软件工程，也可以简单改造后用于多种琐碎而重要的协作中。但是对于很多传统企业，市面上缺少针对性强的现成产品，而这些企业自身也没有精力和意愿，非常深入的做一款适用于本领域管理工具。毕竟这种工具只有一两人用的意义不大，这个组织都用起来才最有意义。

PS：我看燃尽图好像是根据ticket数量的历史变化情况，线性的预测未来的工作进展。但工作真实进展很可能不是线性的，这是否说明燃尽图的剩余工作预测存在天然偏差呢？

作者回复：其实很多工具的定制能力都非常强，例如Jira，可以考虑基于它定制，尤其是可以开发插件，开发成本不高，但可以做很多个性定制化的事。

燃尽图是有天然偏差的，因为任务的复杂度其实不一样的，有的几小时就完了，有的得好几天，有时候你看只剩下一个任务了，但这个可能是最难耗时最长的。

所以我个人更喜欢看板视图，可以直观看到当前Sprint具体什么任务还没完成。



👍 3



小老鼠

2019-09-11

- 1、如何管理好成员学习新技术？新工具？
- 2、如何确定ticket的状态，比如完成，是真完成了还是假消息：-(
- 3、项目经理更重要的工作是什么？

作者回复：1. 我觉得学习新技术和使用新工具是要把握好度的，一方面不能过于保守，排斥新技术新工具，另一方面不能过于追新技术新工具，对新技术新工具不要急于应用在实际项目中，需要小心论证。

相应的对于成员学习新技术，也一样要鼓励他们去学习，将学到的新技术和新工具在内部进行分享。

同时也要限制和约束对新技术的使用，建立合理的机制去验证和推广新技术。

2. 谁提Ticket谁验收验证

3. 项目经理最主要工作就是协调好项目资源，有计划有顺序的推进项目，保障好项目的正常运行。



👍 2



alva_xu
2019-04-01

ms-project这样的计划工具，适合于项目整体计划的把控，人财物的协调。

ticket系统适合于每个阶段任务的安排、变更和任务跟踪。

两者一个全局一个局部，在敏捷项目里应该结合起来使用会比较好。项目整体计划抓大的WBS，不作过度深入的WBS，而ticket系统可以跟踪管理局部的变更，是计划管理的子集。

所以我的经验往往是先做一个全面的迭代计划（用甘特图），基于此做人员安排和工作安排，并拿此作为汇报的依据向领导汇报。

当然，这种模式适用于项目整体目标清晰，时间节点容易规划、每一阶段工作都容易估算的项目。

作者回复: 👍很好的经验分享



👍 2



泡泡龙
2019-03-28

现在正在学习使用码云企业版自带的任务管理，我认为这个软件最大的优点就是1)买了企业版就自带了 2)可以和git联合使用，可以指定任务相关代码库、分支

作者回复: 我没有用过码云的，但Github自带的也很好用。



👍 2



dancer
2019-03-28

jira 禅道都用过，比较喜欢用jira的看板。另外燃尽图不错，做事会有成就感！

作者回复: 用了看板我对燃尽图就懒得看了，毕竟看ToDo栏还有多少Ticket也是很直观的：)

共 2 条评论 >

👍 2



bearlu
2019-03-28

老师，看你这个课程。我刚好遇到类似问题，我们公司做C++静态代码检查是这样的流程。大家上传代码，然后我用PVS检查，然后查出问题，找到对应的提交代码的人。现在领导叫我，写些工具能自动化找到对应提交代码的人，但是我觉得这样不合理，我觉得提交前不合

理就不然提交，我也这样和领导提过，但是领导说这个成本很大？我现在不知道怎么做，听他说做个工具，还是要坚持提交前做检查。

作者回复：我没做过C++的CI集成，理论上是没问题的，例如：

<https://www.viva64.com/en/m/0005/>

<https://www.viva64.com/en/b/0393/>

即使不用CI，也还是可以用一些自动化脚本辅助，例如git hook。

这种事情就属于磨刀不误砍柴工，第一次肯定要投入一些时间精力的，后面会获益良多。具体怎么做还得你自己权衡，也不用一步到位，一点点慢慢改进。



👍 2



busyStone

2019-03-28

请问新的这些工具还能看到并方便的编辑任务依赖么？

有没有工具可以直接通过修改状态就自动换看板的？

另外，像同一个需求需要多端，安卓，苹果，PC同时开发的，请问有没有好的方法来建立任务？之前都是一样建一个，有点烦。

多谢！

作者回复：以Jira为例：

Ticket之间是可以建立关联的，好像不是强依赖。

修改Sprint属性就可以切换看板，修改状态就可以切换看板的泳道。

Ticket可以克隆（Clone），同一个需求可以克隆多份，然后稍作修改。



👍 2



Felix

2019-03-28

工作以来一直用的jira，之前经常使用看板，去年开始我们转用仪表盘了，它可以利用jira中自带的小工具个性化定制想要的东西，公司用jira的同学可以一试

作者回复：我个人更喜欢看板，因为我更关注具体的任务人不是数字图表。



👍 2



西西弗与卡夫卡

2019-03-28

正好和项目经理讨论这个话题。因为项目涉众比较多，如何及时高效披露项目路线图和进展就成了问题。就有同事拿着时间表跑过来过来问，你们进度是什么，一看是去年已经完成的计划。这不是同事的问题。主动披露项目进展、资源使用情况，是项目组的责任之一。有时候需要汇报，但按时凑齐人不容易，如果是多方汇报，那就更耗费时间。

比较好的实践是给项目设定一个主页，里面包括蓝图、进展、资源使用以及问题等。关心的人可以自己订阅。

再好一点的方法是将上述实践做成流程和工具，以便每个项目和团队不用为采用什么样的协同机制而太操心，更多精力放在业务之上

作者回复：你说的几种方法我都试过，用下来发现还是看板最最好用，每天大家根据看板做事情，项目进展和要做什么一目了然。

很多项目管理软件都支持看板视图的



👍 2



Geek_long

2019-03-28

公司用的jira。

作者回复：Jira很好用的



👍 2



hua168

2019-03-28

菜鸟打卡，公司用的是Redmine和禅道

作者回复：赞，Redmine也有听说过，只是没用过。



👍 2



GeeStu

2019-04-05

公司的项目管理工具是禅道，结合钉钉的开发接口做了很多机器人，集成到了公司的自动化部署工具上，每次自动部署结束和任务发布都会在群里@相应的人通知。项目上主要使用迭代开发，两周一个迭代，看起还是很便捷且灵活性也较高。但有时候需求跟任务发布和文档编写的进度相差太大，有时候根本没时间在禅道上去写相关的需求文档，以至于开发完了，测试需要不断跟产品去对需求细节，有时候开发写完了以为完成的相应的功能，但最后发现要做的是A但搞出来的是D，再改成C最后交付了B。所以在这样的交付场景下，需求变更频繁时候如何去平衡使用工具一步一步来和直接口头说明的开发先行文档随后的尴尬关系呢

作者回复: 可以参考用户故事的做法，虽然需求描述很简单，但是有相应的验收条件。

参考: <https://gitbook.cn/books/59d883784730b27bd0d228cf/index.html>

共 2 条评论 >



1



oldlee

2019-03-31

请问前后端开发分离工具有没有好产品推荐？现在遇到的问题是，客户端经常需要等待服务端开发完，才能调用接口联调。

作者回复: 这个有几种解决方案：

1. 服务端先实现一个模拟的接口
2. 客户端自己模拟接口
3. 第三方服务，例如：

<https://github.com/easy-mock/easy-mock>

<https://getman.cn/mock/>

<https://apizza.net/>



1



一路向北

2019-03-29

工欲善其事，必先利其器。

使用好项目中每个阶段的工具，能够提升项目的效率。选择合适的工具和正确的使用工具是其中的关键。之前我们用禅道来做敏捷开发的管理，后来是团队变小了，就没有用下去。现在没有用工具，看完老师的文章，需要重新用起来，用到工具成为长在自己身上一样了，自然效率提高了。

作者回复: 即使小, 也应该用起来的, 强烈建议用起来。

现在很多在线的项目管理工具都不贵, 也不用自己维护, 特别适合小团队。



👍 1



-Helen怪物

2019-03-29

一直在学习工程思想和项目思维, 我是做测试的, 今天的课程让我突然想到可以创建一个质量保证的项目督促团队更好地保证质量, 然后用敏捷思维确保此项目的不断完善和提高, 谢谢! 很棒!

作者回复: 心动不如行动, 加油💪



👍 1