

# 如何从0开始敏捷转型

## 目的

本文希望给想了解敏捷的读者一点启发，并和对敏捷感兴趣的朋友一起探讨、聊一聊敏捷实践带给我们的价值和收获。所以本文以我辅导的两个敏捷转型团队为例子，来分享我对敏捷理论和实践的若干看法，以及如何从0开始敏捷转型的落地过程。

对我本人和辅导的团队而言，是非常受益于敏捷实践方法的。我们的工作成绩可以立刻被看见，工作效率得以大幅提升，成就感加强，我辅导的团队也通过132次的测试环境发布和52次的生产环境发布，证明了团队的合作成果和效率，团队内很多成员在年末都得到了公司的肯定，其中的一个敏捷转型试点项目的方式还顺利通过了 CMMI5 级复评，这更加正向肯定了我们敏捷实践的效果。

## 序言

# GitChat

当今是知识经济时代、信息技术飞速发展和全球经济一体化的时代，对于生活在知识经济时代的我们，现在是最好的时代，也是最坏的时代。一方面我们可以享受到前所未有的全球化个性化服务，比如出国旅游都可以网上预订个房间，从国外买个心仪的产品；另一方面又能感受到动荡无常、复杂模糊、竞争加速的乌卡时代氛围（乌卡是 VUCA 的中文翻译，代表易变、复杂、模糊、不确定的含义，VUCA 是“Volatility-易变性、Uncertainty-不确定性、Complexity-复杂性、Ambiguity-模糊性”这四个词的缩写）。



就在2018年2月23日，又一家曾是全球手机霸主的手机系统——黑莓手机 BlackBerry OS 正式宣布“死亡”，比尔盖茨曾说“微软离破产永远只有十八个月”，任正非说“华为离破产永远只有30天”，如今两家牛逼的公司都在敏捷转型，并华丽转身，都在适应时代和市场的快速变化。所以正如国内的知名精益专家何勉老师在《精益产品开发原则、方法与实施》一书所说的，现在已经不是考虑是否应该敏捷转型了，而是要考虑如何转型的时候了。

虽然标志着敏捷方法论诞生的敏捷宣言在2001年已经形成，但是由于其体系紧贴敏捷价值观、核心原则，并认为敏捷实践永远是在不断变化和持续更新的，所以敏捷转型之风犹如刮过大地的春风，每年都在一遍又一遍地传播到全世界的高科技企业，尤其是互联网公司，甚至许多传统行业的软件部门业主动寻求变身。据统计，大多数公司采用敏捷的主要原因包括：

- 加快产品上市时间；
- 更轻松高效地处理不断变化的优先事项；
- 更好地使 IT 目标与业务目标保持一致；
- 提高整体生产力。

尽管15%的公司报告说他们的敏捷项目都没有失败，但那些确实经历过失败的企业引用了诸如公司理念/文化与敏捷价值核心之间不协调的理由，遵循传统瀑布流程的压力，组织或沟通问题，而且对敏捷实施方法缺乏经验。

在敏捷转型过程中，对于习惯了工业时代的管理方式的团队，也许一下子难以适应这种转变方式，因为敏捷转型会增加工作量，甚至还会减少某些权益，故而会阻力重重；有的公司虽然领导非常重视，但是开展方式不太恰当，一下子铺开，并没有认识到团队人的因素，认为是可以完全复制的工作方式，导致压力重重，效果不好，团队凝聚力下降，敏捷转型似是而非；有的即使进行了敏捷转型，但团队工作方式依然是完全命令式的工作传导方式，层层传递导致业务信息失真；有的敏捷转型时求效果心切，希望越快越好，结果却事与愿违，结果是没有梳理

好团队工作流程，并养成习惯，导致越快越乱；还有的只注重敏捷的仪式，不积累团队转型效果的数据，得不到领导的支持，看不到团队转型的效果，结果又打回了原型，上线时依然要全员一起加个好几天班才能完成，甚至比较长的一段时间，每人每天都需要加班，周末也要加班，使家庭关系和健康都受损，工作效率因工作时间延长而降低，结果是造成人才流失严重，这样长时间的“死亡行军”，损伤士气，最终公司和个人都双输。

敏捷转型其实不是那么容易，也没有那么难，但过程中可能需要多一点点耐心和信念和坚持，并需要辅导团队进行敏捷转型的敏捷教练和团队成员充分认可敏捷实践方法（团队成员也可以逐步认识到这一点），如果能充分理解敏捷，在达到目标的道路上即使走了点弯路，也是对团队和个人实践经验的积累，也更是一种宝贵的财富经历（部分内容来自敏捷教练、创新教练姜信宝的课程《敏捷提高团队效率？》）。



## 一起拥抱敏捷

“Congratulations!”随着老外评估师一脸笑容、挥舞着 CMMI5 的证书高兴地向我们祝贺时，我才恍然意识到我们的敏捷转型试点项目也成功拿到 CMMI5 的认证了，也跟着欢呼起来。我们的评估师还说，还有3年再过认证时我们将拿到的是 CMMI V2.0 的认证，不过由于我们目前的敏捷实践过程扎实，并且有一个与之对应的强大工具平台，已经足以满足2.0版本认证95%的覆盖率。

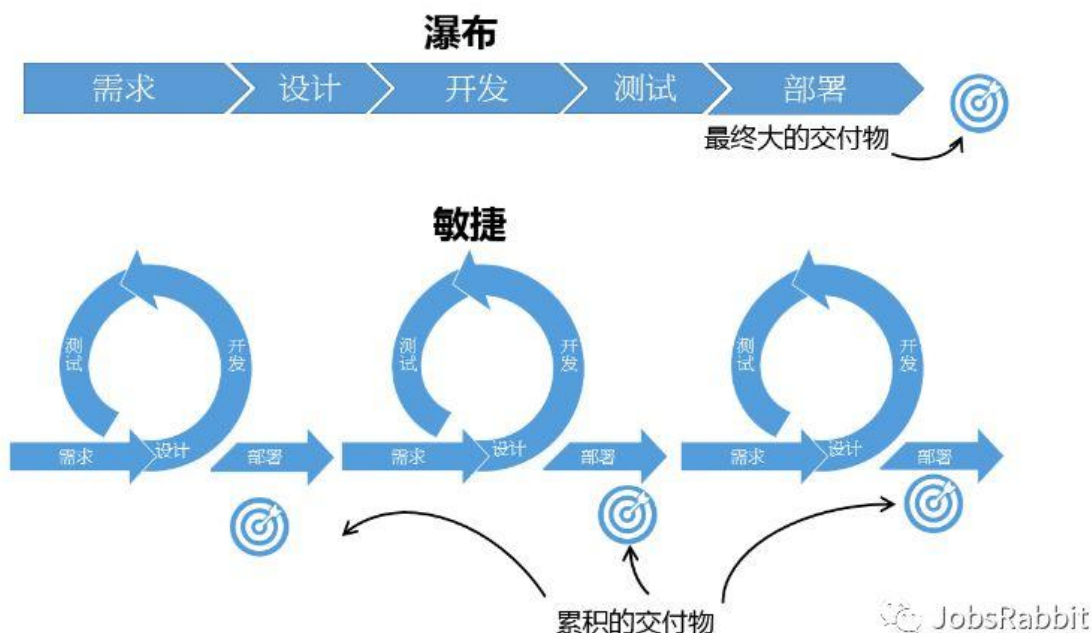
回想在这个过程中，我们采用的 Scrum、Kanban、DevOps 等实践方法已经在日常工作中要求了很多细节，并且是符合团队利益需求和项目交付需要的。作为一名敏捷实践者，我早就期待这天的到来了，一个可以证明 CMMI 和敏捷是可以融合的一天了。我不再是为了过 CMMI 而做 CMMI，而是可以作为一个兼职的团队级敏捷教练，一起和开发团队从我们的项目实践中进行学习和提升，从需求分析、设计原型、开发、测试到部署，充分地在每个过程去了解、实践，去解决每天的问题，每天都是繁忙而紧凑的，因为我们的工作内容和业务环境已经不允许我们像传统的瀑布法那样：执着于先制定一个“完美”的计划，还要等某个功能模块的需求全部分析完并写入需求规格说明书，然后再分给设计人员写入设计说明书，并把 word 文档里的每个标点符号都写得非常正确后，才到开发人员那里进行开发，只有等各个功能模块的开发都做完了并集成后，才开始一轮轮的测



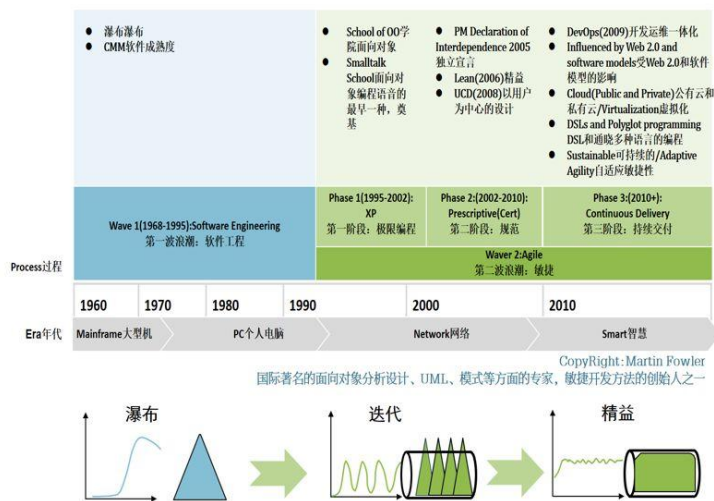
试，测试通过后，才部署到各个环境中再测试，最后才上生产线。正如下图所示的瀑布开发的过程，会导致业务需求严重失真，并且交付到客户手中的产品时间也比较漫长。



在2016年考取 PMI-ACP 认证（ACP 是 PMI 美国项目管理协会在全球191个国家和地区范围内，发起的针对敏捷实践者的资格认证）之前，我和许多对敏捷理解不够深的朋友一样，认为下面这副图就完全阐述了瀑布与敏捷的区别，因为光从交付时间和交付的角度来看，有人说“敏捷就是迭代，就是小瀑布嘛”，因为与瀑布型流程比较，敏捷型在短周期都可以交付成果，无需到项目结束阶段才能交付客户的价值。



可是到了如今的精益开发——后敏捷时代，上面这幅图已不足以说明敏捷开发的快速变化了，敏捷开发的方式已经开始从“迭代式开发”向“流式开发”演进，见下图（部分内容来自华为精益看板专家王明兰的分享《华为软件开发云大规模精益开发实践》）。



#### • 软件危机及软件工程的出现

速度是企业竞争致胜的关键因素，软件项目的最大挑战在于

- ✓ 一方面要应付变动中的需求
  - ✓ 一方面要在紧缩的时程内完成项目
- 传统的软件工程难以满足这些要求

所以软件团队除了在技术上必须日益精进，更需要运用有效的开发流程，以确保团队能够发挥更大效率。这正是Agile Process (敏捷的软件开发流程)于近年来兴起的主要原因。

#### • 技术的更新换代决定了软件工程方法的演进

敏捷开发实践方法体系下最基本的方法是Scrum和Kanban方法:

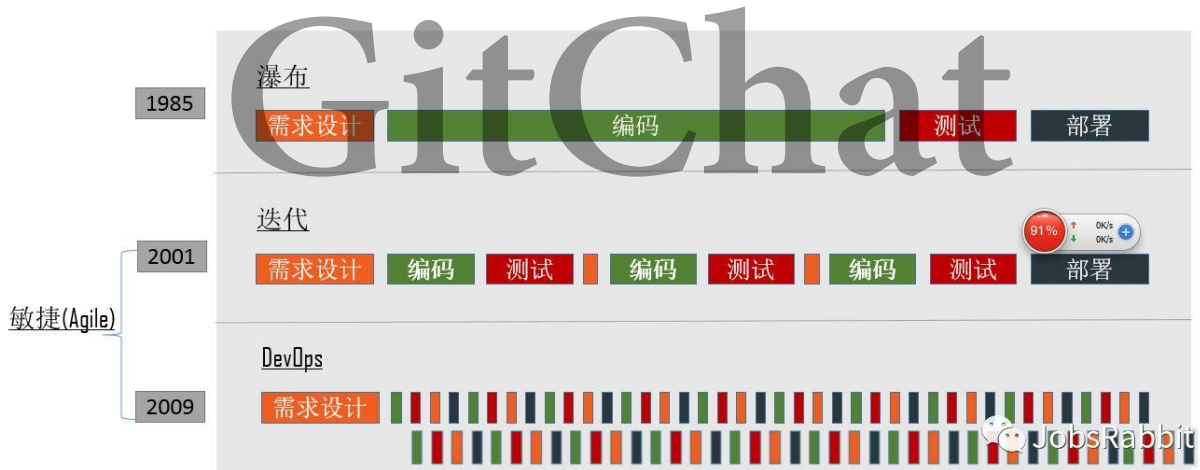
- ✓ 时间盒式: Scrum方法, 注重短周期节奏
- ✓ 流式(精益): Kanban方法, 注重可视化和在制品限制



因此，如果一定要从需求、设计、开发、测试、部署这几个阶段来看瀑布与敏捷的区别的话，下面这幅图才能更贴切地去形容目前我们采用的敏捷实践方法，与之前的敏捷方法相比，我们日常的工作在逐个分解后，可以一项项地有序地并行，从提高了生产率和工作效率；并行的工作离不开工具平台的支撑，在后文的实践介绍中我会介绍在工具平台中，团队是如何并行工作的。

### 项目执行方法的演进

- PROJECT EXECUTION METHODOLOGIES - THE CHANGE



然而要做到这样的地步，一是要让研发能更好地创新、理解业务，二是团队需要采用好、适合自己团队的敏捷方法，三是技术和相关的平台要有保障，因为我们的世界变得越来越快了，不仅业务变化快，技术更新也变得越来越快。正如精益看板专家何勉老师说的“现在是VUCA时代，我们面对新常态，人们不再怀疑敏捷的必要性，而且要求的更多；产品的持续创新事关生死，产品开发部门不应该再被看成组织内部的成本中心，而是要成为价值探索、发现、创造和验证的创新中心，是企业的核心竞争力所在。”

## 充分理解敏捷实践方法

在《敏捷项目管理基础知识与应用实务》一书中，许秀敏博士说“敏捷不全是新的概念，许多概念是从20世纪60年代演进至今，是从软件开发以来，为了解决无法如期交付符合

客户需求的问题所演进而来的。敏捷宣言将这些概念，加入敏捷价值观与敏捷实践等新的概念，整合并记录下来，形成所谓的敏捷宣言，改变了开发团队的思维和心态，让开发团队能完成为客户创造最高价值的软件（Alistair Cockburn,2009）”

所以敏捷的核心既是敏捷的心态，也是一种基于敏捷价值观、原则和实践的做事的方法体系，更是一整套关于价值观、原则和实践的集合，并随着行业技术的出陈推新，在实践层的技术层会有很多变化，但万变不离其宗，所以业界总结出敏捷“洋葱圈”模型（来自光华国际副总裁、企业转型顾问、规模化敏捷专家李建昊老师），分别是敏捷心态、价值观、原则和实践。



在洋葱圈最外面是我们用得最多的敏捷实践，使用敏捷方法论的团队在任何时候都要有开放（Open）的心态和学习精神，因为敏捷项目管理是比较注重实践和变化的，它有核心的价值和重要的原则（可参见《敏捷宣言》），却没有边界，就类似太阳系一样，只要是符合核心价值和原则的实践，都可以被吸纳进来作为敏捷的实践；但如果随着时间的推移，某些实践并未能如它初期那样比较靠近敏捷倡导的核心价值和原则，就会不再作为常用实践来进行推广，比如 DSDM、Crytsal 等方法；而随着敏捷实践逐渐丰富，有一些新的符合价值的实践也可以被吸纳进来，比如看板、用户故事地图，其他类中的“影响地图”、“DevOps”、“容器”、“微服务”等（如下图）。



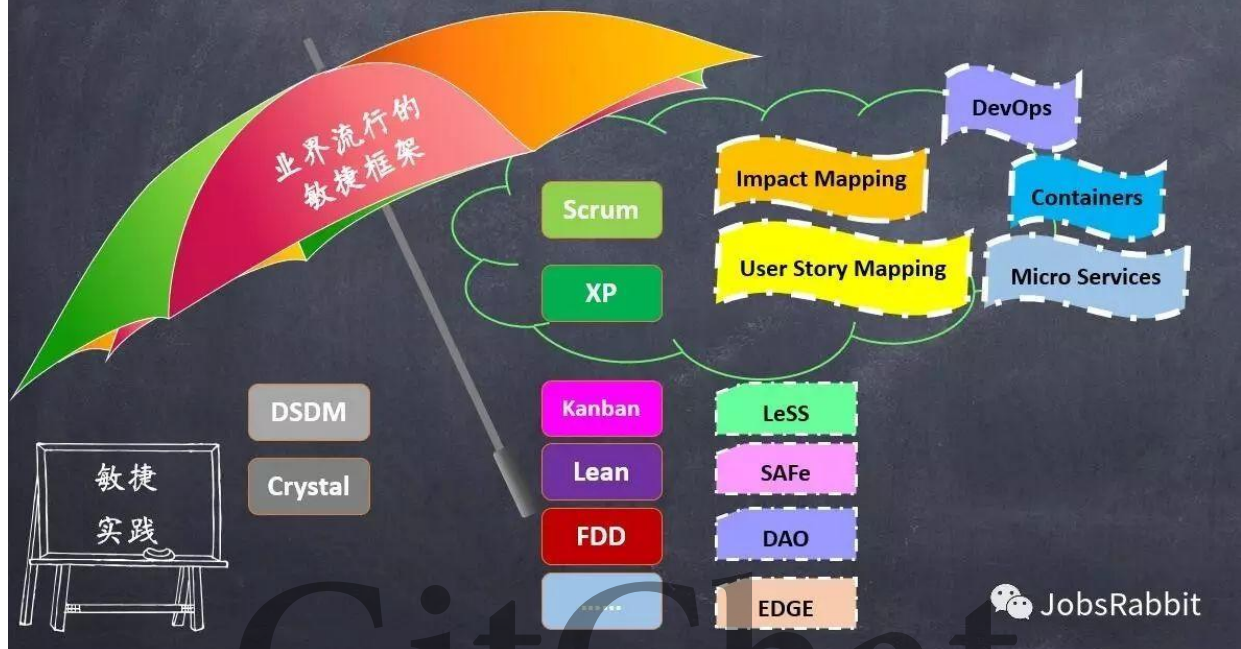
# 敏捷开发实践框架

敏捷实践框架分类：

轻量级：Scrum 、XP

重量级：DSDM、FDD、Lean、Kanban、Crystal、.....

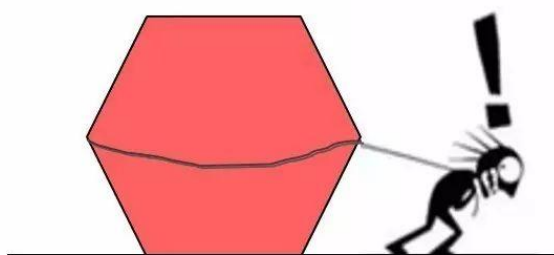
每种方法都有自己的约束条件，要达到敏捷的效果，就要尽量遵守这些约束。



## 如何开始

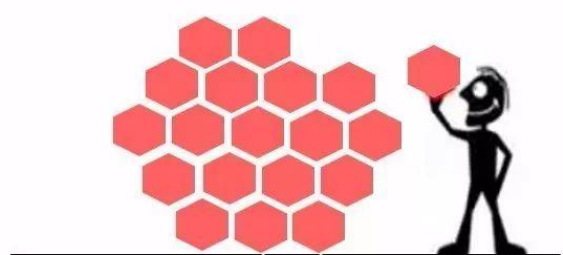
作为目前互联网行业最流行的软件开发方法，敏捷开发方法论的确有很多比传统瀑布型开发方法更为高效的地方，但如何在一个团队里实践好敏捷开发方法是个难点。

瀑布项目过程



“这个项目如此大，  
我不确定能否交付得了它！”

敏捷项目过程



“把项目拆成刚刚好的部分  
进行交付则容易得多呀~”

(翻译自MedDigita的《传统型项目经理可以转型成为ScrumMaster吗?》)

我个人认为前面所介绍到的 ACP 理论的包容性很大，理解前面的内容就可以有更大的包容心和好奇心去学习和实践新的方法，并加以提升，因为敏捷的方法本来就是一门实践

的科学，不是所有的方法都是完美无缺的，都是适合于各种环境和背景的，因此需要我们始终保持着空杯的心态，接受新的事物，并进行实践。这个理论模型认为不是所有的方法都是完美无缺的，都是适合于各种环境和背景的，因此需要我们始终保持着空杯的心态，接受新的事物，并进行实践，考察它是否符合敏捷的核心价值观、原则。所以随着时间的推移，以前的一些敏捷实践不再流行，而一些新的、更符合现代业务环境的新实践方法被吸纳进来，比如从2009年后才火起来的 Kanban 方法，还有目前比较流行的容器、微服务技术等等。

总之，在各类敏捷实践方法中，Scrum 和 Kanban 方法是比较基础的敏捷方法论，下图是微软首席讲师、资深系统架构师李智桦分享过的各类常用敏捷实践方法的约束数目比较图，其中 Scrum、Kanban 是原则较少、最适合团队进行敏捷转型的入门方法的不二之选。



关于Scrum、Kanban方法网上和各类书籍都有介绍，笔者就不在这里赘述；对于敏捷转型而言，Scrum方法结合Kanban方法是较为有效的方法，对于这两种方法，京东首席敏捷创新教练王立杰说“Scrum方法是打造敏捷团队基本方法”，另一位资深敏捷教练也说过类似的话“如果没有实践过一百轮的Scrum迭代，团队是掌握不到敏捷实践的精髓的”，而作为可视化效果最好的Kanban方法则是配合其他方法的最佳灵药，并且也是团队进行敏



敏捷转型的基础实践之一，二者的关系类似于筷子与刀叉，各有优劣。

## 方法与工具



### 适合敏捷转型的业务场景

市场环境：客户首选项和解决方案选项经常变化。

客户参与度：密切协作和快速反馈是可行的；随着流程的进展，客户更清楚他们想要什么。

创新型：问题很复杂，解决方案是未知的，范围没有明确定义。产品规格可能会更改。创新突破和上市时间非常重要；跨职能协作至关重要。

工作模块化：

- 增量开发具有价值，客户可以使用它们。
- 工作可以分解成多个部分，并以快速的迭代周期进行。
- 后期更改是可管理的。

临时错误的影响：它们提供有价值的学习。

### 第一个敏捷转型团队的实践过程

我辅导的第一个敏捷转型团队是在我通过了 PMI-ACP 认证（ACP 是 PMI 美国项目管理协会在全球191个国家和地区范围内，发起的针对敏捷实践者的资格认证）后，充分理解和学习到了敏捷实践的方式，开始在团队内进行试点，约定了团队协作的方式，并逐步改进各个细小的环节，并持之以恒，使团队养成习惯，如下图。

## 敏捷开发过程公约

团队公约：  
● 每轮迭代前共同预习阅读。  
● 团队高度自治。



### 敏捷开发过程公约：

约定团队的协作规则，结合固定时间盒式(Scrum方法)以及流式(Kanban方法)开发节奏形成团队的工作流程，使团队开发活动兼顾计划性与灵活性。

### 敏捷开发过程的主要活动包括：

#### 一、迭代计划会议

产品负责人从日常工作梳理好的“Product Backlog”里挑选最高优先级的部分讲解给团队，团队就需求细化、完成标准询问、逐条估算放入本迭代的开发任务中，直至任务饱和。

#### 二、每日例会&每晚日报

每日例会：每早工作前，团队会进行一个简单的站立会议(15分钟内)，沟通当前进度、下一步任务和当前存在的问题，以借助团队的力量解决。每晚日报：每晚下班前，团队成员会在系统上更新今天所完成任务的耗时和工作状态，便于第二天例会的沟通和进展了解。

#### 三、迭代评审

在每个迭代的最后一天，团队召集评审会，邀请产品负责人等参加，对已经完成的产品功能条目进行评审，后者做出判断并给出改进反馈。

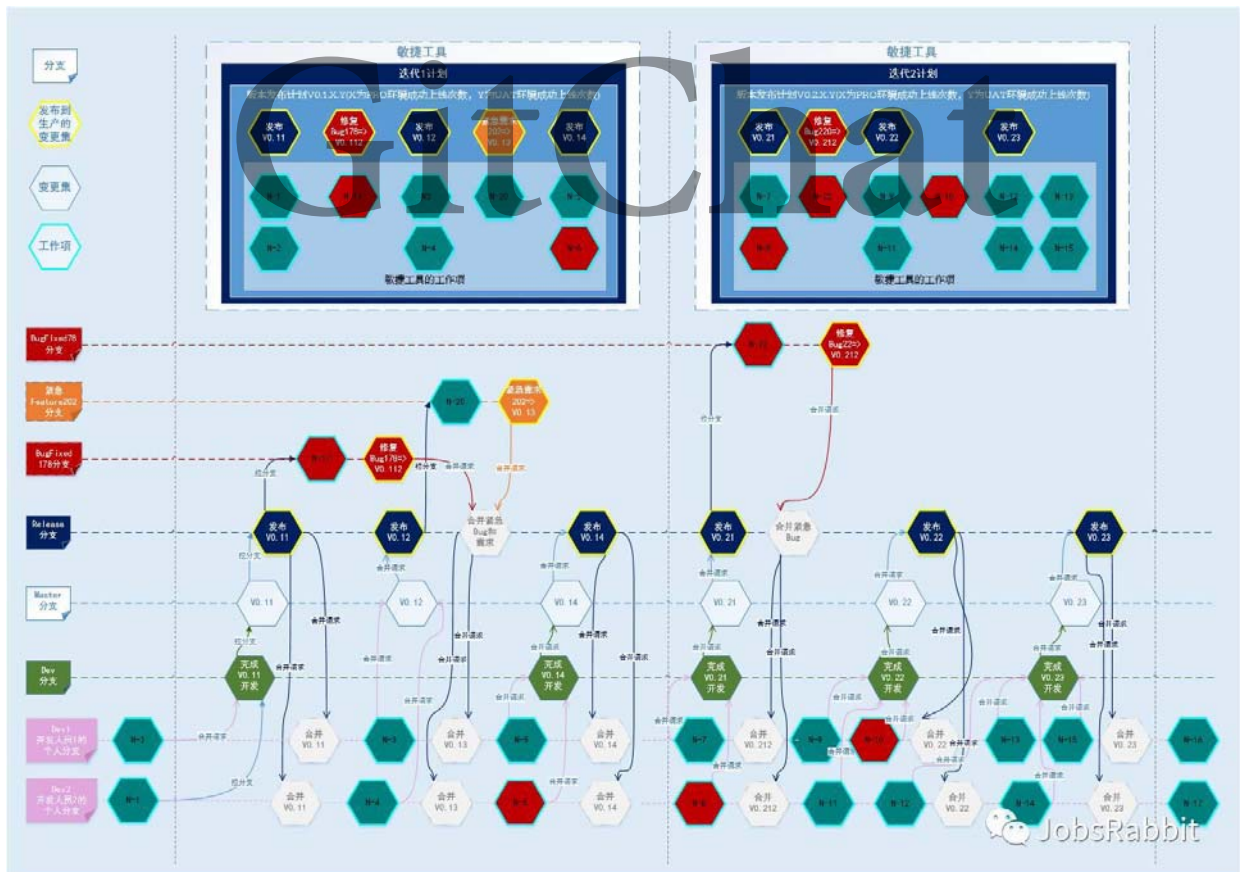
#### 四、迭代回顾

评审会结束后还召开反思回顾会，对本次迭代中的成功与失败之处做出总结，并在以后迭代中进行改进。

#### 五、产品待办会议

产品负责人或产品小组要经常梳理(产品待办会议)可以在迭代中的任何时间段多次进行，一般至少要梳理出下次迭代的产品待办，为下个迭代计划做准备。

在这个团队中，我们的项目业务是有些模糊的，需要快速上线，而且需要配合我们自己的部署平台进行项目交付的发布，也比较符合敏捷实践的背景。我作为兼职敏捷教练，不仅在团队内身兼数职，还可以和团队一起把控和逐步细化部分需求，并管理生产环节的部署，所以我本人是非常关心上线成功率和速度的，通过工具平台规划需求，然后整个项目的需求和代码都是一一对应关联的，从而可视化整个项目的上线过程，见下图描述的团队的“代码发版流程”，并可以实现“迭代计划与代码集的关联”。



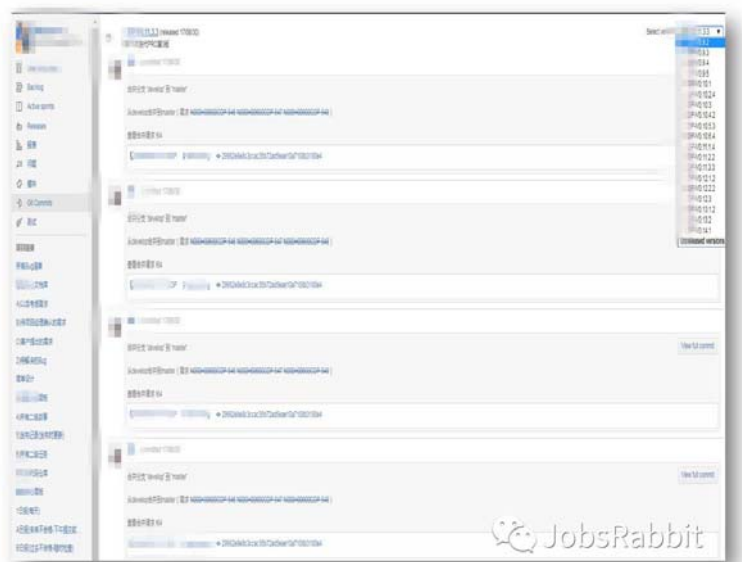
我们的团队从2016年年底起至今，一共走了23个 Scrum 迭代，发布了个132测试版本，52个生产版本，基本上是两周一次迭代，其中每个迭代随需发布多次版本，从各个版本都可以从需求追溯到代码的变化情况，具体参见我们项目库在“工具平台”的版本数据展示图。

## 版本与需求

Releases

版本	状态	日期	开发日期	发布日期	备注
1.0.1	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.2	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.3	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.4	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.5	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.6	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.7	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.8	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.9	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.10	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.11	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.12	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.13	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.14	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.15	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.16	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.17	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.18	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.19	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.20	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.21	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.22	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.23	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.24	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.25	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.26	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.27	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.28	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.29	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.30	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.31	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.32	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.33	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.34	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.35	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.36	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.37	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.38	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.39	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.40	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.41	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.42	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.43	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.44	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.45	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.46	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.47	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.48	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布
1.0.49	已发布	2018-01-12	2018-01-12	2018-01-12	首次发布

## 版本与代码

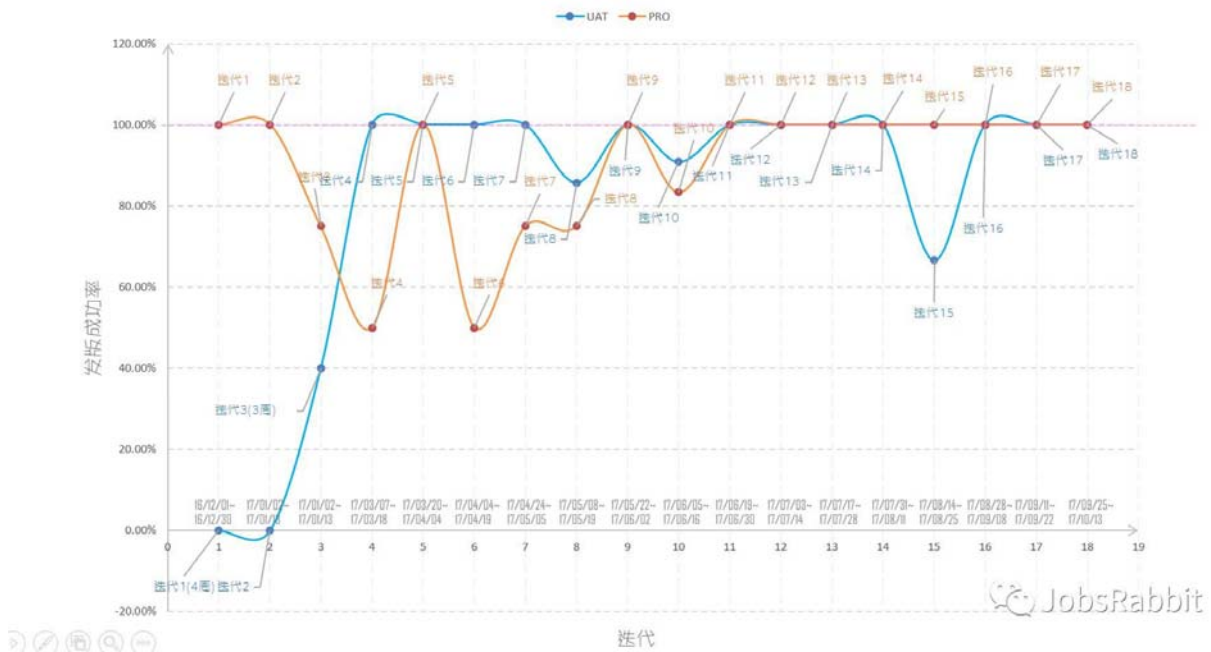


在刚实施敏捷 Scrum 迭代时，我们开始不是特别有经验，最早是4周一个迭代才发布一个生产版本，后两个迭代周期降为3周，发现我们的能力还能提升，最终稳定在2周一个迭代，并且在迭代内可以发布多次版本，发版能力逐渐增强，并且发版成功率趋于往100%成功率稳定（我约定了 A、B、C 三种成功率，A是上线没有 Bug；B是有轻量 Bug，无需回滚；C是上线不成功，或有严重 Bug 需回退版本），参见下图。





UAT/PRO环境的  
各迭代总版本(2周)的发版成功率



在我们的内部平台上提供了用户故事地图、用户故事、Backlog、迭代计划、看板、报表、在线文档、敏捷知识库、即时通讯集成等功能，我们使用工具平台进行敏捷项目管理的主要步骤如下（此部分细节在我们的工具平台上，形成了详细的“在线帮助手册”，供使用平台的团队阅读使用）。

（1）在项目的“过程管理”的“用户故事地图（User Story Mapping）”先添加大需求故事，如下图所示。

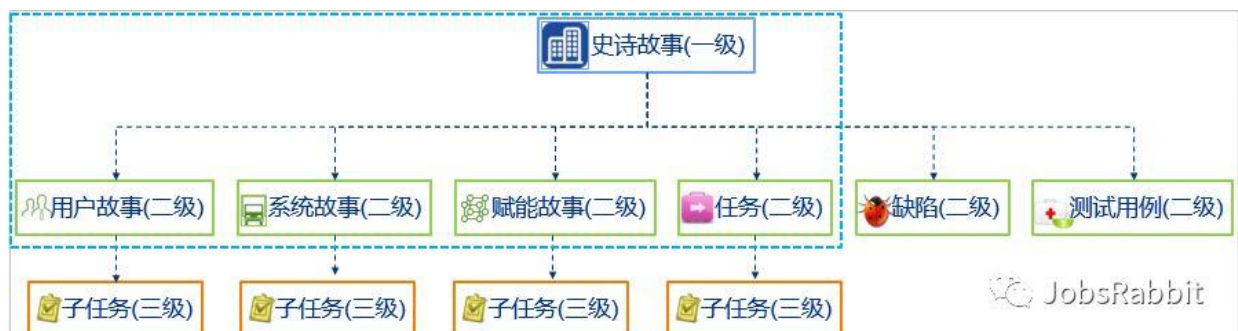
#### ✓ 用户故事地图



#### ✓ Scrum/Kanban



（2）然后将大需求故事（即系统的“一级工作项类型”）拆分为可由团队开发的3种用户故事:用户故事、系统故事、赋能故事（即系统的“二级工作项类型”），并放入发布计划或迭代计划中，如下图所示。



(3) 在《发布计划》或者《迭代计划》中对上面的“用户故事”进行优先级排序（即使用系统进行“需求梳理”）。

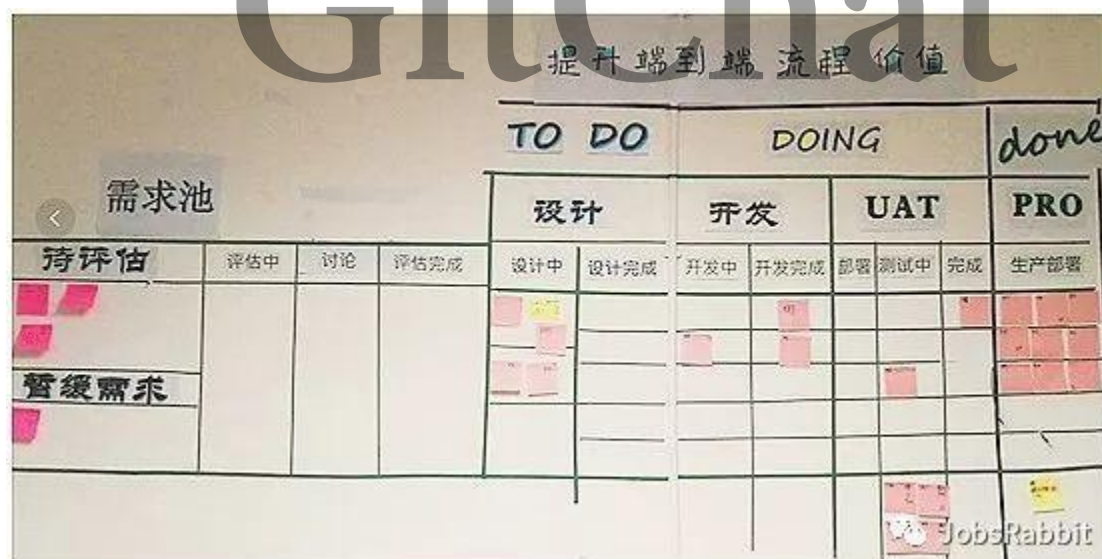
(4) 在《发布计划》或者《迭代计划》中对上面的“用户故事”进行故事估算（即在系统上进行“估算子需求”）。

(5) 在《每日工作跟踪》中对本迭代的故事进行任务拆解，在任务创建和每日工作结束时，录入和更新任务到期日、估算时间和剩余时间（即在系统上登记和录入每日的工作内容、时间记录等基本信息，强制必填）。

(6) 迭代结束后，通过系统演示进行迭代评审，确认本轮迭代的工作都已完成，在“版本管理”页面发布本迭代的所有工作（迭代末的工作登记也要记录到系统上，由系统的报表功能出具相关数据图表，从而使团队进展情况可视化）。

(7) 在对一轮迭代回顾时，可通过点击报表和项目迭代数据面板、迭代需求数据面板和缺陷数据面板可以查看本轮迭代的数据和图表进行分析，然后将本轮迭代的经验教训记录到《项目文档库的迭代回顾》中，最后可通过文档库的会议纪要转化为行动项，进入下一轮迭代。

在每次迭代中，我们还采用电子 Kanban 和物理 Kanban 结合的方式进行结合，全范围可视化迭代工作，其中电子 Kanban 有 WIP 限制的提醒功能；团队面对面沟通交流时，采用物理 Kanban（如下图所示），沟通完毕后，将具体信息登记到系统的电子 Kanban 中，并进行更新操作。



使用 Kanban 的好处除了可视化工作，另一个优点是不同角色的工作可以并行，当开发人员在迭代内的某部分需求进行开发时，业务方（兼测试）可以对已经开发完的需求进行测试环境的部署和验证等工作，正如前面所说的提高了迭代内的工作效率。

在一个迭代结束末，进行及时迭代回顾，并将经验教训形成记录和下一轮的迭代行动计划，这样我们每个迭代的工作都会比前一个迭代更有改善，持续改进。例如，在几轮迭代后，我们对核心的功能做了接口自动化测试，这样对于一些基本功能就无需再次人工测试；在开发人员不忙的时候进行代码的重构，减轻技术债务。最终，从整个团队的迭代执行情况的累计流图可以看出，从我们开始稳定执行2周的 Scrum 迭代工作后，基本

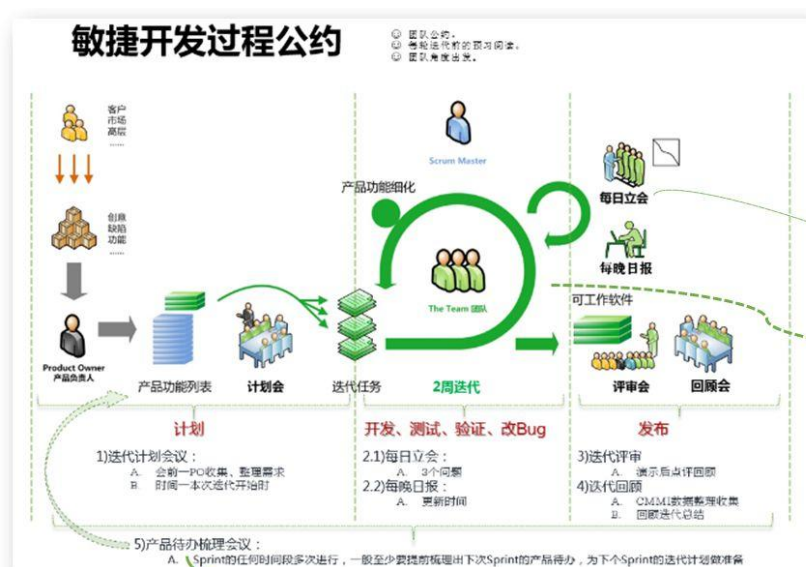
工作处于比较流式的快速开发、快速上线中，这样为我们将来做进一步的流式并行工作打下比较好的协作基础。

## 第二个敏捷转型团队的实践要点

与第一个团队不一样，第二个团队采用微服务技术，天然更快速，但他们会偶尔存在多个项目并行的情况，即某个月突然有个临时极短项目同时在进行。最开始我也想和辅导第一个团队一样，在 Scrum 方式下，采用平台工具结合物理 Kanban 的方式来进行，后来发现物理 Kanban 难以执行，于是先只用平台工具的电子 Kanban，让团队改变只通过即时聊天工具记录需求和 Bug 的习惯，让团队的过程数据记录在平台里，并辅导团队其中的一个技术负责人为兼职敏捷教练，辅助负责迭代相关的工作，比如迭代任务的梳理、迭代中记录迭代的发版情况，迭代末关闭迭代，并邮件截图迭代进展数据面板、迭代需求数据面板、缺陷数据面板给全体成员，使全员对自己所参与的项目的进展有全方位的了解和跟踪。当然这样的实践方式还不只有这些，还有这个团队自身的微服务技术的演进，对我们的容器平台的部署方式的改进也提升了他们的工作效率，从而可以兼顾多个项目。

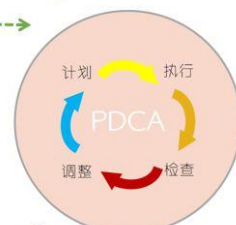
## 结尾

回想起这一年多来的敏捷转型历程，从陌生、忐忑、团队冲突爆发、融合熟悉到默契配合，历历在目。总而言之，是基于敏捷理论上实践好了 Scrum 和 Kanban 方法，并结合工具平台养成习惯，固定周期形成一个个的 PDCA 的改进循环过程，从而提升了团队效率。



敏捷开发过程公约：

约定团队的协作规则，在固定时间盒周期(譬如2周内)各个角色要一起协作、执行的工作，是三个PDCA循环的过程，是不断有反馈、改进的过程。



JobsRabbit

对于我个人而言，也是工作成效非常大的一年，在这一年中，除了我日常的一些工具平台的运维工作以及 EPG（过程改进小组）的若干工作，我还在两个项目里担任了敏捷教练，并亲自负责和操练其中一个项目的需求、设计、测试到部署的整个过程，从而可以帮助改进过程，最终辅导团队过了级；在这一年我编写了很多材料，虽然压力很大，收获也不小，但敏捷实践的修炼是无止境的，我希望将来有更多的收获和大家一起分享。