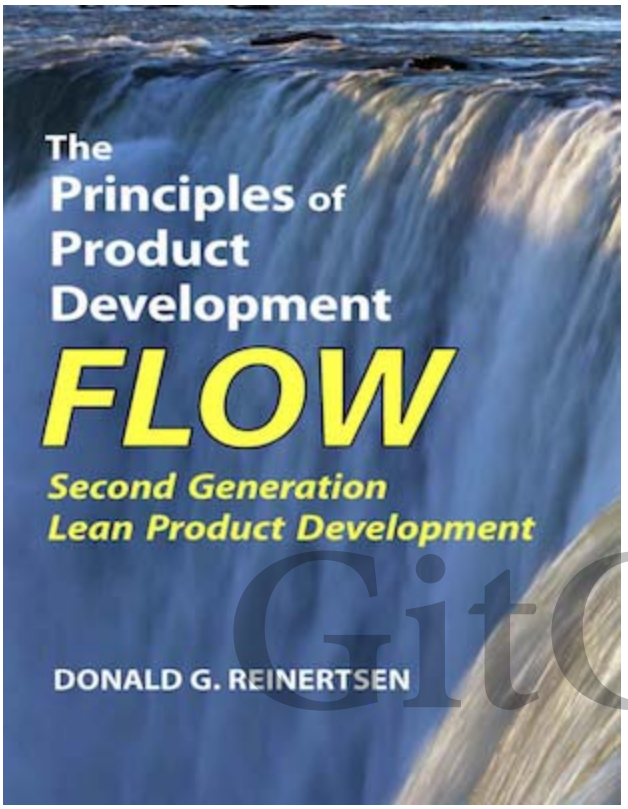
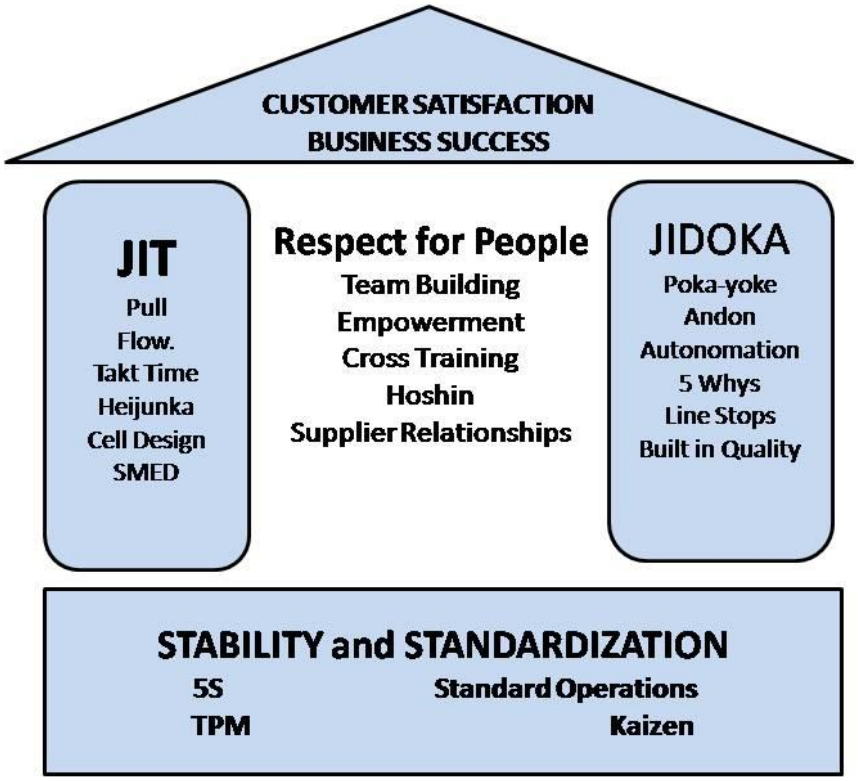
组织实施精益的五大误区

从2007年开始，业界都在尝试将精益生产理念引入到软件开发管理，比较知名的包括看板方法和精益产品开发原则。



十年过去了，精益一词越来越五花八门，包括精益创业、精益数据分析、精益扩张……但在软件开发方法上面，似乎没有过多的突破。其中的重要原因，正是因为通过看板和一些开发原则，过度放大了精益工具的作用，而忽略了背后的思想。毫不夸张地说，过度强调**看板和流动**，恰恰步入了“缓解症状，但不解决根本问题”的反精益模式。除此之外，还有一些“看起来很像精益”的工作方法，比如说持续交付流水线，小批量提交代码，就是更狭窄的视角了，因为我们甚至说不清楚什么时候提交代码是最合理的，看板也不会告诉你今天提交多少代码比较合理。因为持续交付的目标是**发布可靠的软件**，背后借鉴了精益生产中的自动化（Jidoka）理念：Poka-yoke、Andon、Autonomation、Line Stops、Built in Quality 等。



如上图所示，看板和持续交付分别部分实践了精益中两根支柱 JIT 和 Jidoka，这就好比单个棋子无法完成一盘棋局一样，我们要谈的正是如何步出这种“棋子思维”的误区。

误区一：流动之前忽略了稳定化

许多组织引入看板，解决的最大问题却并不是“准时交付”，而是作为一个轻量级的管理工具使信息透明。信息不透明并不是缺乏看板工具，而是团队协同不够好：瀑布项目中的里程碑和任务管理就足以透明项目和工作状态了。

作为轻量级管理工具的“看板”，并不会给团队带来任何精益的工作方式。于是我们看到各类的阻塞和风险、不能按时交付。

精益生产中实施看板之前需要先进行生产平准化，包括标准作业、缩小批量和快速换模实施。对软件开发的借鉴意义在于：

- 标准作业：尽可能保证在进入开发前后的任务卡片数量一致。这意味着所有的故事卡、任务卡应该在开发之前有效分解，在开发过程中不断向后流动，而不是到具体某个环节和角色处才开始进行分解。
- 缩小批量：缩小批量的目的恰恰不是为了降低并行，而是为了更好地有序并行。小批量的版本移交有助于开发测试之间的连续流动，也有助于多个项目之间的任务快速切换。换而言之，小批量的发布是一种方法，是为了提升团队对变化的适应性，而不是因为适应能力提升的结果。

> 什么现象是不稳定的？
>
> 整体负荷过重：看板堆积过多的卡片，还未流动或发生较多状态改变，就持续增加很多新的卡片。
>
> 交付节奏不稳定：在各个泳道上分布不均匀，或经常出现大量移交后等待。
>
> 可用资源不足：反映在具体某个人或某个环节积压卡片严重。

如果我们知道“平准化”是好的，而非仅仅从工具上控制“流动”的效率；那我们就可以从现象出发找到和解决根本问题：

- 促进更小的任务拆分，直到负荷稳定化；
- 促进产能合理分配，直到产出稳定化；
- 促进消除技能差异，直到交付稳定化。

误区二：限制 WIP 忽略了综观全局

限制 WIP 的目的是什么？是为了缩短开发周期时间（Cycle Time）和提升开发可用率（Availability）！所以，当最终团队的产能、响应临时任务的能力没有提升时，WIP 的设置一定存在问题。

> 团队 WIP vs. 个人 WIP
>
> 发现很多组织将 WIP 设置在个人身上（通常 <= 3），这个数字实际上等于团队中每个人满负荷，并没有起到限制 WIP 的作用。
>
> Henrik 在《精益开发实战》中的实践是“每次团队挑选出优先级最高的 8 个功能卡”，这其实就是团队 WIP 限制。
>
> 将 WIP 设置在团队，才能有效控制所有人的工作总量，也有助于团队共同推动整体的进度。而通常的情况是，许多卡片都需要两个或更多开发人员的合作才能完成，所以为单个开发人员设置 WIP 其实增加了任务安排不协调的几率。比如说，一个开发人员正在完成另一个开发人员的依赖任务，突然有其它任务加入，根据 WIP 设置他可能会接收这个任务，但无意中另一个开发人员的任务就被阻塞了。设置 WIP 在团队，这个任务就需要在团队层面进行评估，是否接纳，即使接受后也需要决定开发优先级。

限定团队 WIP 的另一个好处在于，根据利特尔法则：Throughput = WIP / Cycle Time，我们可以评估团队的生产效率，对积压需求进行交付预测和资源调配；另外也可以针对 Cycle Time 进行持续地改进以提升生产效率，这也符合在精益中“消除变量”的改进思想。

误区三：闭门画流程忽略了现场改善

精益的一个基本假设是：一线人员的知识总和超过所有的专家。真正的精益改进是在一线进行，改善是一种现地现物的活动，并且充分发挥开发人员的才智。

在一家组织中发现精益实施竟然是请来咨询师进行研发管理流程设计，这无疑是完全不懂得精益的思想。

> 改善的原则 Kaizen
>
> 精益中的改善不是像在回顾会上列出几个问题，或是开始一些行动，它有标准原则。
>
> 第一，不要解决现象（Don’t fix symptoms）。
>
> 第二，不要停止找到根本原因（Don’t stop at the first root cause）。
>
> 第三，使用目标—资源—约束框架找到改变行为的方案（Use the goals-resources-constraints tool to address behavior-driven causes）。

改善是一种深思熟虑，**精益不是也永远不可能是过程管理或研发管理流程**，而是通过提供一系列的工具，触动组织成员变得善于思考。

除改善之外，价值流设计提供了组织层面的顶层改进设计。它的基本步骤是：

1. 识别价值创造：理想状态
2. 设计价值流：未来状态
3. 识别差距：转变

改善和价值流设计，分别提供了短期和长期的组织改进工具。

误区四：集成产品开发等同于集中式组织结构

与敏捷的全功能团队相反，精益产品开发的模式不但不是全功能团队，而是通过一体化（Intergration）整合人员、流程和技术，发挥出巨大的杠杆效应。换句话说，精益是一种高效的经营管理模式，而非强调组织结构效率。在复杂业务的组织中，更不可能实现固定组织结构的效率，而在于人员的协同能力。以下是我在《金融组织敏捷的逻辑》一文中列举的功能型、产品型及项目型组织的优劣势。

基于功能结构的专业化组织

如：机械工程部、电气工程部、制造工程部……

优势：

- 使用同一种专业语言，沟通高效；
- 利于加强专业技能的深度；
- 利于将方法和技能标准化；
- 利于按项目需要有效部署和充分利用人员。

劣势：

- 容易倾向于自己的专业，衡量成功的标准是部门绩效和获得的预算，而不是企业、产品以及客户的成功；
- “只有自己才是救世主”，在协调配合方面，没有哪一个部门能做得特别好。

基于产品结构的专业化组织

专注于一个项目或产品，设置清晰的目标，从所需的功能部门抽取代表，参与产品开发和流程。

优势：

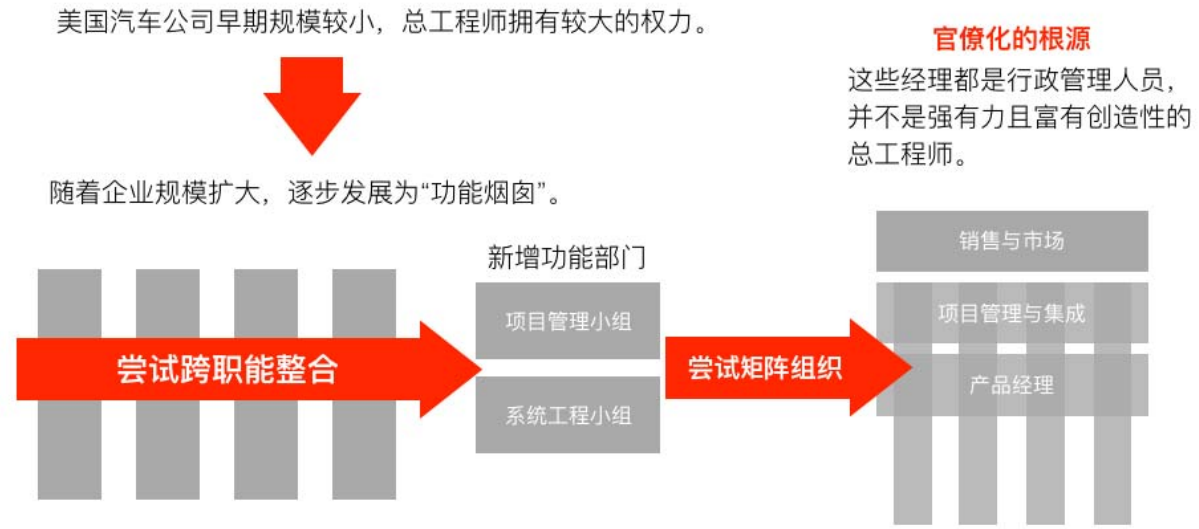
- 围绕共同的目标工作，创造满足客户需求的产品；
- 有效沟通合作以缩短开发周期；
- 从多个角度制定产品和流程决策，并确保决策能够达成共识，从而提高产品质量。

劣势：

- 产品功能的标准化丢了；
- 各产品或项目交叉资源利用率极低；
- 衡量团队成功的标准变成了对人力资源的争夺和产品或项目预算的争夺。

项目管理或产品管理的专业化组织

另一种常见形态是将项目管理和系统集成组成专门部门。当组织需要面向客户和产品交付时，又组建一批产品经理部门。但正如上面所述，这都终将走向官僚化组织。因为，它使关键人才的活动违背了“现地现物”这一核心思想。



误区五：产品开发能力等同于营销能力

从早期一些传统企业步入电商，以及对数字营销大力投入，到现在对“流量”的焦虑，这是片面地理解了用户，也片面地理解了产品。流量对互联网公司来说是一个影响可直观控制的变量，整个互联网的架构就在于“抗流量”，而对于传统企业，尤其是银行业来说，流量的增长引发的是多米诺骨牌效应，后端系统性能、业务复杂度、支撑人员，将呈几何级数增长；没有整体的运营规划和价值流设计，流量无异于一场洪灾。

精益改进致力于帮助组织找到真正的价值并消除浪费。对金融组织而言，有价值的尝试在于：

- 建立以客户为中心的系统和流程：金融产品流程长，客户接触环节多，反而从体验和数据上难以保持一致性。保险行业更早地打通了单一客户视图，不少国外保险公司早就在尝试实验式营销，在银行业还比较少见。
- 全渠道集成：包括全渠道绩效和归因，现在各个流程的渠道对接非常分散，疲于“转发”流量，但很难度量绩效。
- 整合前后台产品流程：在金融行业并没有一个高效的产品开发过程，这导致信息丢失和后端部门疲于奔命。这也应该是精益最大发挥作用的地方，创造有价值的产品，并减少过程中的浪费。

对过去的银行运营绩效而言，通常会考察每 FTE（Full Time Equivalents）的客户数量和帐户数量，而当银行走向开放以及多样的资产结构时，精益通过明确价值、保持小批量地交付，可以帮助整个组织低风险地尝试创新。