# 敏捷开发绩效管理之生产率度量方法

<http://blog.csdn.net/column/details/agiledeptchen.html>

度量敏捷开发的生产率一直是个难题，确切说度量任何开发方法的生产率都是一个难题，但它实际上有答案，这个答案是本文的主要内容。

## 度量敏捷生产率的目的

真正难以回答的是度量生产率的目的是什么？

很多人都认为是考核绩效，发奖金。根据上一篇文章的内容我们可以知道，这完全是行不通的：客户并不购买我们的生产率，生产率高也并不能证明产品或项目盈利，应该为团队设立外部目标，否则很可能得到一个生产率很高，但是实际上很烂的产品——质量上或易用性上很差，抑或其他想象不到的，但一定遇到的原因。这是我们说为什么用外部目标而非内部目标考核团队的原因。

或许又有人说：“开发得快不快是团队的事情，产品好不好则是产品负责人的事情” 。这样也不对，相当于我们在自组织之后，当我们享有勇气、尊重、沟通……这些敏捷的特质之后，我们居然得到了一个只关心自己的开发速度，而置客户价值和企业利益于不顾的团队。“受激励的个体”只被自己小团队的绩效所激励，并不爱也不关心自己的产品，这绝对不是敏捷开发的初衷。

**度量生产率的目的不是绩效考核，而是希望提升生产率绩效。将度量结果进行横向和纵向比较，可以分析造成生产率差异的原因，并进而提升生产率绩效。**

## 微观度量方法-故事点

### 故事点估算方法

“每月完成的人天数”这个方法不说了，用尺子量尺子，肯定不行的。不过通过统计每月的实际投入天数，可以优化人员利用率，并间接提升生产率。

“每月完成的故事点”这个是比较好的方法。

所谓故事点法，就是提前选择一些大家都熟知的、以往做过的、典型的故事，比如：

1. 对单个表进行增删改查

2. 对父子表的增删改查

3. 为一个已经存在的数据表增加一个复杂报表

4. 修改一个中等难度的BUG

……

（实际上这些故事应该是具体的，而不是像上面例子中一样看起来更像是“分类”）

然后人拿出当年的历史记录，将当时所投入的人天数称为“故事点数”（也有别的做法但这个最简单）。比如“对单个表进行增删改查”当年用了4天，那么标准故事点就是4。

当下次估算时，人们又发现有一个故事也是“对单个表的增删改查”，于是就先选定基数为4，再讨论这次与上次比，到底复杂多少。如果一致认为可能复杂20%，那么故事点就是5。

如果大家的生产率不变，那么这个故事应该5天完成，但是如果结果却是4天就完成了，则表明大家的生产率提高了。当然不是一个一个故事度量，而是把整个迭代内的故事点加起来度量。

通过**纵向比较故事点，可以知道大家是否比以前的生产率提高了**。

横向比较故事点比较有难度，因为每个团队乃至项目都会选定自己特有的标准故事，而且极难说明这个团队和那个团队的标准故事的转换关系。

### 故事点的局限性

在推广故事点这件事情上笔者有所保留，建议尝试但需注意风险，必要时知难而退。在笔者遇到的这么多做敏捷的企业和人里边，还没有见到有人提到他们的故事点应用是成功的。

原因在于找到大家都熟知的、以往做过的、典型的故事很难，而让所有人记住它们当年的详细情况以便日后对比修订就更难。

04年笔者去做咨询的一家企业有他们的故事点模板（他们并不做敏捷开发，但却使用完全类似的方法），一共有17种标准故事，已经记录了25个项目的故事点数据和实际工作量数据，每个项目从4个故事到上百的故事不等。他们希望笔者能帮他们计算一下“17个标准故事分别对应多少人天”。终于遇到了又有标准故事又有历史数据的情况，这比所有一穷二白想使用故事点的企业乐观多了。

这是一个所谓“线性规划”的问题，涉及“最小二乘法”、“超越方程”这些玄乎的名词，但却在Excel表里有这个功能，不过是10分钟的事情并不费力，真正费力的是解释其结果：求解的答案是——某些标准故事是负数，也就是说如果把这几种故事当作负数对待，那么以往发生的25个项目的预测结果与实际结果最符合。

再换一种直白的解释：**用这17种故事预测工作量不准。**

或许有人会说他们的17种故事选得不好，或他们的水平很差。怎么说呢，他们是一家1000多人的电信企业，专职做过程管理的人就有5人，还认真地记录了这么多数据，恐怕当年选定故事的过程也是经过思考的。倘若他们都难以建立其故事点，一般的10人团队想自己做一套恐怕更难。

回过头来观察他们公司的**失败原因，是在为新的故事找到对应的标准故事后，没有根据其差异进行调整**，而是机械地选择了标准故事的故事点，导致误差很大。在采用故事点的时候应该注意。但他们考虑到某些故事的回归结果居然是负数，即使“进行调整”结果也会是血淋淋的，甚至可以说基本扔到了标准故事从头估算，最终放弃了故事点，采用了另外一种方法，就是下篇文章提到的“功能点估算”。

笔者之前的一篇文章有故事点估算方法的更详尽的介绍，但角度不同：[敏捷估算：故事点与直接估算天数的差异](http://blog.csdn.net/cheny_com/article/details/6223742)

故事点为我们提供了一种比较客观度量敏捷生产率的方法，但其局限性限制了其应用。下一篇文章将介绍另外一种广泛应使用的方法：功能点估算法。

直接估算天数或用故事点估天数，都很“程序员”。如果在项目的早期，面临与客户相关的报价问题，或高层领导要统计公司绩效并想进行项目乃至行业间的比较，这两种方法都很难使用。

敏捷开发内部之所以没有进化出来能做**项目间比较、行业间比较、用于早期报价**的估算方法，是因为敏捷的发明者和后来的实践者多数都不管这些事情。而这三样事情，比天数、故事点，在老板眼中更接近生产率绩效。

这时候就需要功能点估算。

## 功能点估算

### 功能点估算的由来

功能点估算是另外一个世界的事情。每100个懂敏捷的人中，可能才有1个懂CMMI；而懂CMMI的人中，可能才有100个人懂功能点，而100个懂功能点的人中，也只有1个人懂敏捷……这就是三个世界，但每个世界都和敏捷世界一样热闹，一样有可操作的方法，只是互相不通信而已——结果是，**每个世界都不知道别人已经早就解决了自己冥思苦想的问题**。

**用功能点度量软件的目的，是在早期获知软件开发的工作量，进而推算开发成本**。 由于这个目的，使得它实际上是与开发工作量的相关性也最强（远远超过Delphi/代码行/故事点/用例点……多个国家的政府使用此估算采购软件，中国大约2年就后采用），而且居然和用户故事还有很好的对应关系。

功能点本身很复杂，大家可以在网上查到一些资料，这里不多说了。标准功能点分析尤其复杂，有一次有一个欧美发包商来到中国，问“我们现在已经有100多页文档，谁看过之后知道这个项目要多少钱才能开发出来，以及为什么，我们就把这个项目给他”。笔者介入了此事，也知道标准功能点能解决这个问题，但是却不能在给定的时间内完成（只有2天的时间解决此问题，而用功能点至少需要10～15天）——国外ISBSG也转发度量Guru Capers Jones的邮件称“只有极少数的项目采用了功能点估算，因为成本太高”。这件事情促使笔者尝试找到简单的功能点估计方法，直到后来在另外一个世界发现有人早就解决了，大约10年了……那就是NESMA的简化功能点，如果不嫌麻烦请参阅<http://www.nesma.nl/section/fpa/>（点击左边 Advanced 下面的 Early&Fast FPA）。

不过建议直接看下面。下面的概念作了**很大的调整以便于用有限的文字理解**，如果有懂FPA的读者看出破绽不要奇怪（本人是正规做FPA培训和研究的）。

### 什么是简化的功能点估算

在我们的开发工作中一共有两类东西要开发，一种是数据，一种是操作。

所谓数据，就是比如要编写一个CRM，其中有“用户、角色、权限”这三种东西，就是要管理的数据，这里权且记下用户有“3个数据”要管理。

所谓操作，就是对用户，应该有增、删、改、查、加入角色……这些称之为操作，这里权且记下对用户，用户会做“5个操作”。

倘若角色和权限没有操作（虽然这是不可能的），那么在NESMA简化方法中由于每个数据是7点，而每个操作是4点左右，那么就可以算出来一共有：

3 × 7 + 5 × 4 = 21 + 20 = 41点。

ISBSG/IFPUG包括中国的CSBSG等都有**不同行业/不同类型软件的生产率统计**，如果你在中国，用C#或Java开发一个类似OA/CRM这样的业务流转软件，那么生产率大约是9小时/功能点（来自于10多个学员的课后数据），也就是上面那个小软件，要用9×41 = 369小时大约是46人天。

“什么？这点内容我不到一星期就能做完”。是，也不是。这一时间的包含了需求分析/设计/编码/测试/集成/上线部署期间的所有时间，还包括开会讨论的时间，和别的功能联调的时间，培训的时间，修改万恶的Bug的时间，提升性能的时间，改善易用性的时间，上网找图标的时间，上班看博客的时间——总之一个真实项目中可能发生的时间全都平摊在这里。

听起来够简单了，但其实还不够。

谁能拿出2页纸的需求文档（假设昨天老板在酒桌上刚从客户那记下来的），就猜出有多少个操作？而且还不遗漏？增删改查好猜，“加入角色”就不好猜了。

怎么办？请看下文。

## 简化功能点分析：NESMA

### 第一级简化

上次说到只用数据+操作就能准确计算规模，听起来够简单了，但其实还不够。

谁能在刚拿出2页纸的需求文档时（假设昨天老板在酒桌上刚从客户那记下来的），就猜出有多少个操作？而且还不遗漏？增删改查好猜，“加入角色”就不好猜了。

NESMA早就遇到过这个问题了，他们这么解决：通过统计发现每个数据差不多有7个操作，所以刚才咱们找出了3个数据，那么：

3个 × （数据7 + 操作4×7个操作） = 3 × 35 = 105

嘿，把角色和权限的操作问题也给解决了，不用猜了。

如果有几个数据要管理也不知道怎么办？那也太粗糙了吧，再去细化一下吧（别细化报表上有几个按钮，按了按钮后的逻辑是什么……那个和规模无关；只确认是不是数据）。

这样准吗？来自课后的10个数据表明，基于这种功能点而作出的工作量估算与实际项目数据相比，最大误差在正负50%左右（本人手里没有详细数据所以没分析，应该取P50就是中值的误差比较好，可能在30%左右）。虽然听起来误差大得乍舌，但是在手里只有2张纸的时候，已经很准确了。某政府部门的要求是到400%以内都可以接受，因为他们在一个项目的招标中，4个供应商报价最大差别是12倍。

总结一下这一级别的简化方法就是：**每个内部数据35点**。另外如果是外部接口数据（比如要取LDAP），那么每个**外部接口数据15点**。

第一级简化适合**项目合同期/项目初始计划**的早期估算。

### 第二级简化

如果项目已经完成了，不用猜了，就可以数到底有几个操作，不用猜测每个数据有7个了。

不过，到软件界面上面数显的很不专业，**如果我们的史诗故事是基于数据写的，用户故事是基于操作写的，数史诗故事+用户故事不就得了**？比如图中这个：



图中就是刚才提到的用户/角色/权限的局部，一共3个数据（小课本是史诗故事），外加19个操作（蓝色的是用户故事，其中两个“新建+查看”分别代表两个，要×2），加起来是 3 × 7 + 19 \* 4 = 21 + 76 = 97，已经很接近105个了……如果考虑到这个软件还没有编完，我们第一级简化还是蛮准的嘛（归功于NESMA的长期努力）。

完成这些功能需要 105 × 9 = 945 小时 = 118人天。如果我们有一个迭代，正好要完成这些功能并将其部署上线，那么就按118天计算吧（如果只编写出来不测试，大约是70%的工作量）。当然这不是说任何一个人都花相同的时间，而是基于现在业界收集上来的数据，团队人均花费这么多。个体的生产率差异可达5倍，但团队却都差不了太多。

另外单个故事的精度也很差，比如如果某人正好编过某功能，可能一小会就完成了。但是如果很多人编写很多故事，大数定理会起作用。

总结一下这级别的简化方法就是：**每个内部文件按7点（每个外部接口按5点），每个操作按4点，加起来就是功能点**。

这个级别的简化**适合每个迭代确定工作量；还可以在项目完成后，总体计算开发效率作为绩效管理的依据（算是回到绩效管理了）**。

## 遗留问题

正如前言，很多敏捷世界的新鲜事，在别的世界早就不是新鲜事了（当然别的世界也有他们的新鲜事），提出和解决下面这些问题的很多前辈都已经去世了。

本文只是指出有这么一种方法，并非这种方法的操作手册，这种方法还是需要培训的（有现成的）。

1. 就这么简单？

不是，简化的功能点大约需要一整天的培训，后续估算（推导工作量/工期/成本）还需要一整天。里边有很多细节。

复杂的功能点大约需要2～3天培训，另外一般5天左右的指导，如果要CFPS证书还有一个3天左右的考前指导。

2. “每个用户操作 = 一个用户故事？那修Bug怎么办？做小的改动怎么办？提升性能怎么办？”

可以记录成不同类型的故事，但是就别计算功能点了。图中三个绿色箭头，就是三个对原有故事的“增强”，其特点就是无法描述为完整的用户操作。

他们不计数，但是在统计时已经被平摊到计数的故事里边了。

我自己实践的时候发现，有些故事为了开发方便，极有可能包含两个操作（如上面的“角色首页”包含新建和查询两个操作），建议可以当作一个故事，怎么舒服怎么来不要太生硬，但是加个记号知道是2个操作，防止数错。

3. 每个数据都是/才能是史诗故事吗？

不一定，我有一个史诗故事叫做“性能优化”，它就不是，也不计算它。

4. 每个动词都是操作？

不一定。简单而言就是只有“主语是用户”的动词才是操作。

暂时还没有遇到有些Backlog Item不是用户操作但是很想当成用户故事的情况，如果有，可以在史诗故事/用户故事上设置一个字段，如果是才计数。

5. 非功能性需求怎么办？

最早提出来也最早被解决的问题，标准功能点的非功能性调整因子是1970年左右提出的（天啊……），有点老了，多数都不适用；个人最喜欢的是韩国标准中的调整因子，大致可以理解为普通的乘1，科学计算之类的乘1.3左右，运营计费乘1.5左右，生死攸关的乘2左右。

非功能性调整因子一般包括规模因子、领域因子（韩国标准包括8大类约50小类）、质量因子（韩国标准包括4个质量因子，每个三个等级）、开发语言因子这几个。有时候不会觉得他们考虑不周，反而是多虑了……

6. 准吗？

不准。因为这种方法虽然很准，但是那个“9小时/功能点”是业界数据，最好用自己的数据重新回归一下。

本人正在回归自己的，可惜只有一个项目在做，所以不得不每个迭代都当作项目来看待。一般稍微大点的企业有一年时间积累20个（子）项目数据就可以了。

7. 有前途吗？

有。芬兰，澳大利亚，韩国，日本……的政府采购均采用这种方法计算价格。中国的国标预计在明年出台，即政府采购项目均将使用这种方法报价（或指导报价）。

8. 有的项目好坏可不是光看开发的功能多少，还要看创意和……

是的，**用这种方法度量绩效的目的，是为了提升开发绩效，加快开发速度，而不是计算工资奖金或评价项目好坏**。在[之四](http://blog.csdn.net/cheny_com/article/details/6713089)里边已经提到，必须在赚来钱的时候才能发奖金，它是由外部目标驱动的。

在产品开发中，往往收入与功能多少的关系不很直接甚至完全没有关系，但在一些直接由功能点报价的政府项目、外包项目里边，这个可以直接作为外部目标考核。