Atitit it业界与软件界的定律 原则 准则

[1.1. 定律 布鲁克斯定律：      人月=人\*月，月≠人月/人 1](#_Toc20433)

[1.2. 摩尔定律（Moore's Law）、 1](#_Toc27530)

[1.3. 安迪——比尔定律（Andy and Bill's Law） 2](#_Toc11138)

[1.4. 和反摩尔定律（Reverse Moore's Law）。这 2](#_Toc23588)

[1.5. 软件开发定律系列之1:3:9定律 2](#_Toc15053)

[2. 五大定律：软件开发中的时间估算.html 2](#_Toc18068)

[2.1. 估算第一定律：估算纯属浪费 3](#_Toc32684)

[2.2. 估算第二定律：估算不可互换 3](#_Toc1038)

[2.3. 估算第三定律：估算是错的 3](#_Toc22807)

[2.4. 估算第四定律：估算是暂时的 3](#_Toc8157)

[2.5. 估算第五原则：估算是必要的 3](#_Toc29853)

[3. 软件开发中的11个系统思维定律 3](#_Toc19219)

[4. 软件项目开发七大定律 3](#_Toc6591)

[4.1. 1. 1:10:100 定律：需求错误导致的成本是修复程序错误成本的 100 倍。 反 3](#_Toc1794)

[4.2. 1:2 定律：在开发中，每花费 1 美元，在维护中就得花费 2 美元，因此要注意度量改进 维护的度量元。 3](#_Toc1295)

[4.3. 3. Weinberg 可靠性零定律： 3](#_Toc1111)

[4.4. 4. 1:3:9 定律： 4](#_Toc31076)

[4.5. 帕金森定律（Parkinson’sLaw） ：工作总是用完所有可利用的时间。 5](#_Toc25597)

[4.6. 布鲁克斯定律（Brooks’Law） ：人月=人\*月，但是月≠人月/人，投入更多的人到一项 延迟的工作上，可以导致该项工作更加延迟。 5](#_Toc8104)

[4.7. 7. 80-20 定律：80%的错误集中于 20%的模块；80%的错误来自于 20%的人员； 5](#_Toc12344)

[5. 参考 7](#_Toc20994)

## 定律 布鲁克斯定律：      人月=人\*月，月≠人月/人

## 摩尔定律（Moore's Law）、

科技行业流传着很多关于比尔·盖茨的故事，其中一个是他和通用汽车公司老板之间的对话。盖茨说，如果汽车工业能够像计算机领域一样发展，那么今天，买一辆汽车只需要25美元，一升汽油能跑400公里。通用汽车老板反击盖茨的话我们暂且不论，这个故事至少说明计算机和整个IT行业的发展比传统工业要快得多。

人们多次怀疑摩尔定律还能适用多少年，就连摩尔本人一开始也只认为IT领域可以按这么高的速度发展10年，至于以后当时他也说不清了。而事实上，从二战后至今，IT领域的技术进步一直是每一到两年翻一番，至今看不到停下来的迹象。至少，在我第一次刊登这篇博客的2007年到2011年的今天，摩尔定律依然适用。在人类的文明史上，没有任何一个其他行业做到了这一点。因此，IT行业必然有它的特殊性。

## 安迪——比尔定律（Andy and Bill's Law）

## 和反摩尔定律（Reverse Moore's Law）。这

## 软件开发定律系列之1:3:9定律

# 五大定律：软件开发中的时间估算.html

## 估算第一定律：估算纯属浪费

## 估算第二定律：估算不可互换

## 估算第三定律：估算是错的

## 估算第四定律：估算是暂时的

## 估算第五原则：估算是必要的

# 软件开发中的11个系统思维定律

# 软件项目开发七大定律

## 1. 1:10:100 定律：需求错误导致的成本是修复程序错误成本的 100 倍。 反

## 1:2 定律：在开发中，每花费 1 美元，在维护中就得花费 2 美元，因此要注意度量改进 维护的度量元。

反思： ? ? 在我们公司的项目中维护成本与开发成本的比例是多少？ 我们在需求开发、设计过程中为了降低维护的成本采取了哪些措施？

## 3. Weinberg 可靠性零定律：

如果一个系统不要求是可靠的，那么它能够满足任何的其他 目的；换句话说，如果对实际工作的程序没有要求，那么你能满足任何设置的编码交付 期。 反思： ? 在限定了资源， 而项目工期又比较紧张时， 我们通常牺牲了什么？我们是否真的加 快了进度呢？ ? 成功的项目管理追求四要素的平衡，即需求、质量、进度、资源。

## 4. 1:3:9 定律：

随着软件系统规模的增大，其成本成倍增长，呈现 1:3:9 的关系，称之为 软件产业的非规模经济现象，举个例子：制药行业，只要做好配方，一条流水线就可以 生产几吨的成品药，药的成本微乎其微；相较定制化的软件来说，面对的用户越多，系 统功能越多，产生的需求和缺陷就越多，开发成本也成倍数增长。 反思： ? ? ? 5. 我们如何降低软件的开发成本？ 为什么提倡采用迭代的生命周期模型？ 为什么提倡小项目、小团队？

## 帕金森定律（Parkinson’sLaw） ：工作总是用完所有可利用的时间。

如果你给自己安 排了充裕的时间从事一项工作， 你会放慢你的节奏以便用掉所有分配的时间； 容易达到 的目标将使员工工作上变得松懈。 反思： ? ? 如何规避帕金森定律？ 如果整个项目有 20%的缓冲时间，你会如何分配这 20%的缓冲？

## 布鲁克斯定律（Brooks’Law） ：人月=人\*月，但是月≠人月/人，投入更多的人到一项 延迟的工作上，可以导致该项工作更加延迟。

1. Barry Boehm 认为：可以将软件开发进度压缩 25%，但是不能再多了； 200/20/6X 现象：人数增加 1 倍，工期缩短 20%，缺陷增加 6 倍。 反思： ? ? 在实践中，我们是否经常通过给项目组增加人手的方式加快进度？ 有哪些合理的加快进度的措施？

## 7. 80-20 定律：80%的错误集中于 20%的模块；80%的错误来自于 20%的人员；

1. 80%的 错误集中于 20%的类型；80%的软件废品和返工是由 20%的缺陷引起的；80%的资源 是由 20%的构件消耗的；80%的工程活动是通过 20%的工具完成的；80%的进展是 20%的人完成的。 反思： ? 在实践中，我们应该如何运用 80-20 定律？ 每一条定律的反思都值得我们在实际项目管理过程中关注。

# 参考

Atitit 生活总常见定律 原则 准则

Atitit 管理学心里学的50大定律

计算机行业三大发展定律\_百度文库.html

计算机行业的三大发展定律-CSDN.NET.html

五大定律：软件开发中的时间估算.html

软件开发中的11个系统思维定律\_知识库\_博客园.html

浅谈软件开发定律系列之1\_3\_9定律 - 柳记 - 51CTO技术博客.html

软件长寿法则，记住这7条 - 研发管理 - ITeye资讯.html ( ned detail)

软件项目七定律引起的反思\_百度文库.html