几个超炫的专业词汇

从同事的<u>博客</u>上学会了几个超炫的专业词汇,激动不已。觉得这些词汇可以言简意赅的概括我的好几篇博文,自己的文章水准真是自愧不如。现在来见识一下真正大师级的英语词汇:

• Yoda Notation (Yoda 表示法)



在 C/C++ 里面使用这样的表达式顺序:

if ("blue" == theSky) ...

这是为了避免意外的写成:

if (theSky = "blue") ...

"Yoda 表示法"的名字来源于《星球大战》的 Yoda 大师。他说话的单词顺序相当奇特,比如:"Backwards it is, yes!"

同事认为:使用这个表示法是为了"变通"(wordaround) C/C++ 的一个设计抉择:使用 = 来表示赋值,而使用 == 来表示比较。这个设计充分的展现了"先辈的罪"(Sins of our Forefathers)这一词汇的精髓。

关于 Yoda 表示法我有不同的见解,请参考《Yoda 表示法错在哪里》。

• Mental Speedbump (头脑减速杠)



由于设计的不协调性造成的用户的注意力分散。比如,很多软件喜欢弹出一个窗口问你"是否继续?"

• Pearl Effect (珍珠效应)



珍珠是怎么形成的?是由于异物掉进了蛤蚌的外套膜和贝壳之间的夹层里面,没法排出来。异物不断的刺激该处的外套膜,又痒又痛,于是外套膜分泌珍珠质把异物包围起来,包了一层又一层。久而久之,就形成了珍珠。

在软件里面也有很多这样的"珍珠"。由于早期的挠人的设计错误,用户不得不采用一些"变通方

案"(workaround)或者"附加过程",这些就像珍珠质一样。久而久之,这些变通方案凝结起来,变成了"软件珍珠",不了解它们来源的人都视之为宝贝。虽然产生于同样的原理,"软件珍珠"远远没有真正的珍珠那么好看。

(请比较: Sins of our Forefathers)

• Sins of our Forefathers (先辈的罪)



当时看起来合乎逻辑并且合情合理最后回顾起来却很傻b的历史遗留设计。

与"珍珠"相比,这些是有意识的加进去的,而不是不小心造成的,虽然这两者都会造成"变通"(workaround)。

• Katrina Effect (卡特里娜飓风效应)

这个词描述的是一种飓风过后完全重头来过的悲惨景象。这种现象现在经常出现在重装或者升级软件之后,或者 Windows 安装完软件之后要你重启机器(关掉所有窗口)。

• Workaround (变通)



因为开发过程的失败而让用户必须进行的一些操作。这些通常是设计失误。

• Jenga Code

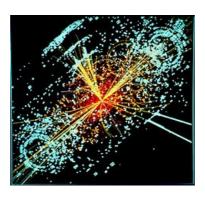


当你加上一小块代码之后,就整个垮掉的那种代码。

Jenga 是一种非常流行的 party 玩具,如图。它的工作原理是,先把那些小木条堆成一个规则的塔。然后,参

加游戏的人轮流从下面抽出一块(只能用一只手)来放在最上面。谁放上之后木塔垮掉了,谁就"胜利"了。之后 这个人就要做其他人想出来的一些"惩罚",跟真心话大冒险那些事情差不多。

• Higgs-Bugson



一种假想中的 bug。它一般是跟据运行日志的少数记录和零星含糊的用户报告推测出来,但是在开发员的机器上很难重现。

• <u>Heisenbug</u>

$$\sigma_x \sigma_p \ge \frac{\hbar}{2},$$

当你试图观察它的时候就突然消失或者改变行为特征的 bug。