所谓"人为错误"

昨天是一个让人悲哀的日子。旧金山湾区主要的上下班交通工具Caltrain,在24小时之内发生三次事故,撞死三人。 其中一次事故发生在Menlo Park,一辆汽车被困在铁轨上,因为被前后的车辆堵塞而无法逃避,终于被飞驰而来的 列车撞成一堆废铁。开车人被消防队员从残骸里切割出来,送往医院后不久死亡。(<u>新闻视频</u>)



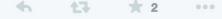


我为生命的殒灭而悲哀,然而让我更加悲哀的是,每当这样的事故发生,总有人指责说是"人为错误"。比如,Twitter上有人说这事故是因为死者没有遵守交通规则,才导致自己的汽车被困在铁轨之上,所以她死的活该。



Joe Burgoon @Keekers16 · 2h

@KTVU why people should never sit on the tracks. People trap themselves by putting themselves in that situation. Just wait. What a shame



真的是因为她不遵守交通规则吗?真的有人愿意把车停在铁轨上等死吗?也许是这规则没法遵守,或者设计得让人很容易"违反"呢?

首先,规则必须要让人理解,切实可行,才能叫做规则。



但是请看看铁轨交叉路口上的指令:"不要停在铁轨上(DO NOT STOP ON TRACKS)","保持路口畅通(KEEP CLEAR)"。我也不想停在铁轨上啊,可是我刚开到铁轨上,前面的车就停下来了,过不去怎么办?另外什么叫做clear?一定要等到路口里面完全没有车才可以进去吗?如果路口里面虽然有车,然而它们都以每小时30英里的速度行驶?这时我还该停下来吗?如果前面车的速度不到每小时5英里呢?如果前面车辆貌似很快,结果我一进路口它就慢下来了怎么办?

如果"不要停在铁轨上"的指令我想遵守都不可能,如果连clear这个单词都定义不清楚,这还叫什么"交通规则"呢?既然规则都不清楚,又怎么能责怪别人不遵守?我要有多么高的预知未来的能力,才能猜得到前面的车会不会正好在我开到铁轨上的时候停下来,把我堵在铁轨上呢?也许你已经看出来了,这其实不是开车人的错误,它最多算一个"判断失误"。每个人都有可能在那种模棱两可的情况下发生判断失误,因为你没法知道前面的车会怎样运动。记者在现场采访的几个开车人都说:"过那个路口要极度小心,因为你不知道前面的车会怎么样走。"

如果你仔细看看卫星图,就会发现铁轨前方的道路狭窄,而且不远处有一个红绿灯。如果这个红绿灯变红,那么就有可能把直到铁轨处的车辆全都叫停。如果你熟悉湾区的道路,就知道红灯处是82号公路(El Camino Real),上那条路的红灯经常等很久。也就是说,可能有很多车在那里等红灯,一直到铁轨的地方!



如果你再仔细一点,用Google Map的street view去实地看一下那个路口,就会发现,地面上的"KEEP CLEAR"字样,其实是用来给被堵在铁轨上的车预留后路的。然后你就发现,如果后面的车不遵守KEEP CLEAR的指令,那么它们就会断掉铁轨上的车的退路。所以,其实不是铁轨上的车自己等死,而是后面那些不遵守KEEP CLEAR指令的车,把它逼上了绝路。然而就像我之前提到的,想要遵守KEEP CLEAR又是很模棱两可的事情,后面的车有可能以为你过得去,所以才跟上的。所以你死了,不能怪火车,不能怪你自己,不能怪前面的车,还不能怪后面的车!怪谁呢?只能怪路口的"设计"!

这种路口交通规则还有一个致命的特征,那就是后果的严重性不明显,人不会敏锐的感觉到犯了错误的结果是车毁人亡。一般人都不闯红灯,因为很显然,如果你红灯不停就会被另一个方向的车撞上。可是违反这铁道路口的规则,后果不是立显的,有可能什么事也没有,也有可能呆在那里几分钟之后才出事,到时候想逃都逃不了。这就像把活青蛙放进冷水里,然后慢火加热一样,它不会立即被烫得跳出来,而会死在里面!等你慢慢的开到铁轨上,才发现前面的车不走了(因为更前面路口亮了红灯),后面的车又抵上来。过一会儿,当当当,栏杆放下来,火车来了……你这是在设陷阱诱捕野生动物吗?

如此容易出现的失误(甚至不叫做自己的失误),真的值得一个人用生命来偿还吗?按照这样的逻辑,我就可以把地雷埋在大街上,插上标志牌说:"下面有地雷,不要踩!"如果你踩了,那我就可以怪你没遵守规则,自己找死!

如果你回头看看历史就会发现,Caltrain几乎每个月都会撞上至少一辆汽车,所以这次的事故绝不是偶然,它有更深层的原因。上一个月,我乘坐的一列Caltrain,就因为前面一趟列车撞上了汽车而延误了好几个小时。当时我就在Twitter上看到有人责备开车的人,说他脑子秀逗了,不该把车停在铁轨上。当时我就在Twitter上警告@Caltrain,说你们应该仔细分析一下这个交叉路口的设计,也许是因为设计有问题。没有人回应我。这次出了三条人命,交叉路口的设计问题才终于受到了重视。

出了人命的大事故,也许能唤醒人们一点理智,认识到所谓的"人为错误",其实在很多时候是设计错误。在这个例子中,交叉路口的设计是不合理的。一旦你因为判断失误把车开进去了,就有可能出现无路可逃,车毁人亡的局面。然而很多生活中的设计失误所引发的"人为错误"都是不致命的,有点像慢性毒药。这种貌似无关痛痒的设计错误,更加容易被忽视,它们就潜伏在我们的身边。

在我所在的软件行业里,就有很多这样的设计错误。在我看来,整个软件行业基本就是建立在一堆堆的设计失误之上。做程序员如此困难和辛苦,大部分原因就是因为软件系统里面积累了大量前人的设计失误,所以我们需要做大量的工作来弥补或者绕过。举个例子,Unix/Linux操作系统就是一个重大的设计失误。Unix系统的命令行,系统API,

各种工具程序,编辑器,程序语言(C,C++等),设计其实都很糟糕。很多工具程序似乎故意设计得晦涩难用,让人摸不着头脑,需要大量时间学习,而且容易出错。出错之后难以发现,难以弥补。

然而一般程序员都没有意识到这里面的设计错误,知道了也不敢指出来,他们反而喜欢显示自己死记硬背得住这些稀奇古怪的规则。这就导致了软件行业的"皇帝的新装现象"——没有人敢说工具的设计有毛病,因为如果你说出来,别人就会认为你在抱怨,那你不是经验不足,就是能力不行。这就像你不敢说皇帝没穿衣服,否则别人就会认为你就是白痴或者不称职的人!Unix系统的同盟者和后裔们(Linux,C语言,Go语言),俨然形成了这样一种霸权,他们鄙视觉得它们难用,质疑它们的设计的人。他们嘲笑这些用户为失败者,即使其实有些"用户"水平比Unix的设计者还要高。久而久之,他们封住了人们的嘴,让人误以为难用的东西就是好的。

我体会很深的一个例子就是Git版本控制工具。有人很把这种东西当回事,引以为豪记得住如何用一些稀奇古怪的Git 命令(比如git rebase, git submodule之类)。好像自己知道了这些就真的是某种专家一样,每当遇到不会用这些命令的人,都在心底默默地鄙视他们。作为一个比Git的作者还要高明的程序员,我却发现自己永远无法记住那些命令。在我看来,这些命令晦涩难懂,很有可能是因为没设计好造成的。因为如果一个东西设计好了,以我的能力是不可能不理解的。可是Linus Torvalds的名气之大,威望之高,有谁敢说:"我就是不会用你设计的破玩意儿!你把我怎么着?"