

IT 沙龙-现代软件工程

Check by SCMChina ++C

2002.10.08 v1.0



8848SOFTWARE-SCMChina

IT 沙龙-现代软件工程

第 4 7 期 I T 沙龙 现代软件工程

主办单位：中关村 I T 协会

承办单位：书生公司

协办单位：新浪网

时间：2 0 0 1 年 1 0 月 2 1 日 下午 1 4 : 0 0 - 1 7 : 3 0

地点：翠宫饭店三层休息厅

记录单位：北京文山会海速记公司

联系电话：1 3 0 0 1 2 8 7 6 3 3 / 6 8 1 5 9 0 7 2

主持人：

大家好，欢迎大家光临第 4 7 期 I T 沙龙，我是 umlchina。本期沙龙主题是“现代软件工程”，由我为大家主持。现在我们请本次 I T 沙龙的组织者，书生科技有限公司董事长王东临先生给大家做一个开场白。

王东临：

大家好。我这期主题是现代软件工程，我们特别重视“软件”这两个字，我们软件经过了几十年的发展，在我们中国目前软件业里面存在一个问题，就是软件行业的发言权并不掌握在软件人手里，对软件的认识还停留在 7、8 0 年代教科书上写的什么生命周期法，从上到下，由粗到精，这种软件工作方法和原则还是正确的，但是从具体方法来说，9 0 年代以来有一些发展和提高了。传统的教课书上的方法在已经条件充足，条件恒定不变，给出一个静态的最优环节应该说还是一个比较恰当的方法。但是应用在商业环境里面，条件还不是很充分，应该说还有一定的缺陷。

我们因为种种原因，我今年是 3 1 岁，我上学学的这套方法，我们很多学者也受这种影响比较大，而咱们国家有一点跟美国不一样，咱们因为历史原因，很多做软件的做高科技的就受科研单位，大专院校的影响比较大，包括企业很多，美国的企业全都是跟科研单位脱勾的没关系的，他是一个研究机构，但企业本身是私营的独立的。而咱们国家不管是中科院还是北大，还是清华，咱们都是依托着科研单位，这是咱们国家历史的情况。那么这种媒体也好，咱们用户也好，特别用户受这种影响。包括政府官员，他们缺乏市场经济商品经济的熏陶相对比较大一些。

包括我本人，我在 1 1 年整个的从事软件工作中，我普遍感到从用户、媒体还是政府也好，大家都还停留在 7、8 0 年代软件工程的水平。这样对行业造成一定的误导，而且这种误导现在

还是有一定的作用，有一定的影响。

那么我们从目前来说，我们软件行业从经营体制，从企业的体制来说，慢慢的和科研单位的关系慢慢的脱离了，不管是联想，当然联想已经不是软件公司，但是从经营体制来说已经是非常独立的。但是从软件工程来说，就存在这种真正充斥在很多媒体上软件工程上的东西，并不是真正做软件的人相对来说少一点。

所以今天希望借这个机会，请真正做软件的，包括摩托罗拉的，包括国内比较优秀的像金山，也有相对比较符合中国国情的，规模比较小的，像做超级解霸的等等，大家谈谈真正的实际软件工程到底怎么回事。

我们的主持人潘加宇他有个人网站，在软件业在专业圈里面影响也比较大，他本身对软件工程也有很多了解。今天很荣幸请到几位主讲嘉宾和主持人，也希望在座的各位在当前的这种情况下，对现在的软件产业如何去发展，就这样一个最新的，当然也包括周老师，周老师虽然也是在高校里面，但是周老师本身对现代软件工程方面也有比较好的研究。所以我们希望今天大家都有很好的收获。

谢谢大家。

主持人：

今天我们非常荣幸邀请到了五位主讲嘉宾，他们分别是微软研发中心服务系统平台部经理杨永生先生，金山公司金山词霸研发部技术总监邹涛先生，摩托罗拉全球电信解决方案部网络方案技术中心高级经理胡大庆先生，清华大学著名教授周之英老师，还有一位是豪杰公司的总裁梁肇新先生。

首先我们有请清华大学计算机系的周之英教授。周教授二十多年来一直从事软件工程方法的教学和研究工作，曾经和主持和参与多项软件工程有关的国家重大科技攻关项目和 8 6 3 项目，还有需要大型复杂网络的应用系统开发工程，连续十多年来在清华讲授研究生课程“软件工程技术和设计”。可能大家对周教授最大的了解就是 9 9 年末的时候出过《现代软件工程》的书，当时涉及的内容是很新的。所以从当时很新的书里面，周老师对新知识的敏锐和软件工程推广的热情。

周老师今天发言的主题是《现代软件工程的观点——尊重事实、认识极端，解决思想》。下面有请周老师。

周之英：

我很高兴有机会跟大家一起探讨现代软件工程。我大概二十多年一直对软件工程很有兴趣，为什么呢？如果软件是一个项目的问题，那软件工程我们希望解决普通性的问题，超越一个项目，所以对这个问题我认为很有挑战性，所以一直在搞这方面的工作。

大概两年前出了一本书叫做《现代软件工程》，为什么加“现代”两个字？主要想界分一下现代软件工程跟软件工程有一些差别，什么是软件工程？那个时候就希望改变小作坊方式的软件开发方式，转向工业化的，这个工业化是从什么角度来看，就是希望学习工厂的经验，也就是物质生产的过程。所以物质生产过程基于一些规律，物理学原理力学等等，这些东西的道理工程的一些方法。所以这样能够使生产工作得到大规模生产，效率很高，不需要很多的记忆，比如一个零件需要很高的手工，现在用了这种工程规范以后，就不需要，就是学一下工程规范就可

以做到了。这是软件工程追求的理想。就是把这个软件变成一个物化的观念。

但是90年代以后我发现了很多矛盾，因为我参与了很多项目，就觉得并不像那个样子，这里面最大的矛盾是什么？软件不是纯物化的东西，这里面有人的因素，所以这就有很多变动的东西不可能像理想的物质生产过程，基于物理学的原理来做的，人的认识规律我们还没有认识地很清楚，所以到现在就有五花八门适应各种情况的东西出来，才能真正解决它的问题，适应变化的需求。所以我基于这样的观点，全球化形式下，变化多端，因为过去是很稳定的情况还可以，现在全世界这么大范围里面，任何一个地方发生变动，就可能影响到你这儿，所以变动是非常地迅速，这时候人的因素就更突出了，而且方法就更多种多样了。所以我就加上一个“现代”，就改变过去软件工程的一成不变的用一个方法解决一切的，而且多种多样的丰富多彩的问题。

到现在为止，我这样的观点，由于书上写的也不是很清楚，匆匆写出来以后，然后加上我的声音有限，跟各方面交流也不够，所以很多的思想也没有得到很充分的发展。这次有机会跟大家谈谈我也觉得很高兴。有很多的东西我个人能跟参加实际交流我觉得可以更好地展开这些思想。

我今天主要想提十个观点。主要是针对沙龙给我的参考文章里面提到的普遍问题。

一个观点，现在的项目各个都不一样，每一个项目都应该有自己的规范，规范都是不一样的，那么软件工程是超越项目。就是说每一个项目不一样，但是项目之间有共性的有规律性的东西，我们要把这些东西研究出来。所以软件工程是近项目产品，不可能脱离项目产品，但要超越产品，超越开发组织，开发的团队，个人，就是要超越它要有客观性，而是要那出事实和东西来，这事实东西就是软件过程中一系列的东西，包括档案等等这些东西。所以所有的工作是要建立在客观性对象基础上。这是第一点，就是说要超越项目，共性跟个性的问题，而这些东西不能评主观想象，而要有一个事实作为基础，要有可能性。

第二个观点，软件工程理论方面，经验实践，各式各样标准等等，是跟物质世界相应的东西是有区别的，就是说软件工程的这些理论、方法、标准、开发模式等等，是强烈依赖于你这样具体的环境跟执行的人，就是说这个人的本身不像一个车间一个机器将这个过程，这个机器生产的東西不变的，但是软件工程在这个过程中，每一个人不可能他就有不一样的。所以对于这样人的能力，人的行为，人的一些东西，比较难以刻画，所以所有理论东西是不充分的，因为人在这其中是起很主要的东西，因为模型总归是个简化，但是人的简化往往是太简化了。所以我们应该看到这些理论到目前为止不充分的。

不充分以后你要去推广就有一定的难度，我们就说所有理论要使用的前提是什么？可重复性，像CMM，最基本的条件是可重复性。但可重复性你的边界条件都是不重复的你那这个东西就不好重复了。所以你要认识哪些东西重复哪些东西不重复了，你要推广就难度比较大，不是不需要推广，难度比较大。这就需要人在里面调整，因为每个人在这里面，实际上有的东西是能推广，它隐含了人在这里面的能动性调整的东西就清楚了。所以你学习的时候也要有这样。第二个这是理论的东西是一个抽象纯化的东西，我们要在实际运用里面得要这把纯变变成实际的情况。这东西实际上我觉得很奇怪，大家都对软件工作者说，你这个东西怎么跟书本上不一样，我们想物理学上书本上说动在恒动，静在衡静，这个要在绝对光滑的水平面上不停地滚动，他是教这个，你从来没有在客观实际上去要求，这球在地毯上滚，一会儿就停住了。所以说这个问题你学到的理论第一个是不完整的，很多边界条件没讲清楚，他也讲不清楚；第二，它在实际上不要求它照着书本上理想模式来范本这个东西，这样要求是不合理的，是需要你体会你的环境，而理解它为什么这样做，对你有什么帮助，而不能希望百分之百的照抄，这样你什么也学不到，而不能学到对你有共性的东西。

第三，对于文档。我们要客观性，这中间有一个很大的东西就是记录文档，通过文档记录一些东西。大家说这文档里面的东西也有错，我要照着它做不是就错了。问题是说文档你知道错你就可以改，如果没有文档你是不是就错的更多，那你就要想记录一些东西，大家交流，你想你的，我想我的，这是一个方面，就是文档起的作用。第二个文档应该记录一下开发的历史，这个历史起什么作用，大家想人类社会的历史起了什么作用，对了解过去，认识现在，认识将来。所以我在这里面更提高一些，就是说文档应该是表示你对问题的理解记录下来，你工作的进展应该记录下来，这是文档的作用；第二个从更深层次它是历史的作用，将来可以改进，记录下你的成就。所以对软件组织或者软件工程师个人来说，如果你有抱负，你希望能起历史作用你就想把你的工作记录下来。所以应该用更高更深层次的程度来观察这些东西，

那么历史有什么作用？我们只能根据历史去预测未来，因为我们的历史并不是我们为了躺在老祖宗身上，而是我们为了向前看，为了现在的工作，为了将来的工作，但是现在工作怎么来看，我们还没有做，将来工作更不知道，那只有什么？只有靠历史，靠历史去预测未来。但是我们不能靠历史去计划未来，我们未来一定是跟历史一样的，中国还是在封建社会这是不可能的，还是在文化大革命这是不可能的，我们是为了更好的，我们希望未来比过去历史更好。所以我们不能照过去怎么现在就怎么样，而是要吸取过来历史经验教训来指导我们的作用，这就是历史的作用。

第五个，什么是历史，历史就是过去的事实，要证实的历史。软件工程是很高的项目的成功的经验一个总结，我们软件工程有一些东西，比如说观点分析更多的是过去成功的经验，那为什么软件工程你要去学习这些东西，这是为了掌握历史的观点。你一个人脱离了历史你就是没有文化的。有人说学了有用没有，当然你要没有历史你可以做一点事，但是往往是短期的。你要有更高的要求的话，不是就事论事的话，就应该学习过去成功的经验，而不仅仅是为了当前这件事，这素养的提高是很重要的。

历史要尊重事实，尊重事实就两方面，一方面是历史，一方面是现状，现状也是一个事实，跟历史不一样的，所以你要正确认识自己和别人的情况。

第七点，在全球化的进程中间技术环境的变化非常多样，而且复杂，而且变化非常迅速。对一个具体个人或者一个企业来说，周围不确定因素太多了，针对这样情况，还有一个从软件工程本身来说，对它的要求实在太高了，为什么呢？他现在使用软件工作已经不像四十年、五十年、七十年是非常少的人群，现在而是非常多的人群，各行各业，现在有很多不是学计算机的，学化学的学机械的，有很多人甚至不是大学毕业的，他也在搞软件，也要搞工程。所以他从软件工程希望的东西太多了，就是非常广泛的这样的情况。广泛了以后希望它是简单好用，还要实用，还要有效，它要处理非常复杂的问题，各式各样的问题，还要给人创造一些就业机会，所以这给软件工程带来了很多的矛盾。根本的矛盾是什么呢？这是我的观点，我认为你开发中间所有的能够看到的東西，跟运行的情况发生了变化，就现在面向对象技术出现了以后，运行视图，运行的描述的东西跟你开发的东西不一样了。从前结构化的时候，还可以，差不多，因为你从结构化的模块运行是一个线，这样的东西，你是怎么运行的，你实际上是什么样的模式，运行也是这样的模式，如果我觉得我希望运行的时候效力提高，那我在什么地方我去改一改，人可以看到的一些东西我能够去做它，看不到的东西全放在运行过程当中，全放在脑袋里面，对人的要求太多了。所以想尽办法也难以上青天，就处于这种状况下就非常困难。

实际上运行模式是什么呢？是硬件确定的。第二个是开发平台确定的。也就是说现在的这种硬件开发平台是适合四十年代五十年代一直到现在的，某种意义上适应一个真正要解决这样开发的很多的难点的话，得从根子上来解决。所以现在还是有很多人在研究就更基础的问题。我们可能从国内这方面考虑的少一点，但是我们应该认识到，你不要企图用什么方法能够去解决那

么困难的问题，他也看不见，即使有那样子的话，是会有困难的。这是有根本的矛盾的困难。

第八个，一般性的问题很难解决，但是具体问题并不难解决。一个一个成功具体的项目很多，那项目为什么成功？就是它因为采用了技术，开发模式等等，就能把它主要问题主要特征解决了。现在就有一些方法很极端，比方有人说XP，或者更极端一些，我什么方法也没有，就完全可以用黑客的方式就可以去解决某种问题了等等，这种东西是强调某个片面，就是强调局部，针对具体问题可能是有效的，或者强调不是那么片面的，但是不是很全面，所以就产品中中间件问题，具体问题具体解决的各式各样的方法，各种情况的多样性。还有一种极端就是想办法，我也碰到，因为我有时候去审查别人论文，或者项目，他就希望搞得非常全面，就说我们现在有一个知识库把所有知识都放进去等等，这样的东西。像这种东西超越了你的控制能力，这种东西都是一种想法，有道理，但是能用吗？你得自己考虑。你应该去理解它的一些内涵的东西，比方说它有一些方法非常全面，把各种东西刻画了，也许对你全部做不合适，但是你可以做一种参考模型，你理解他给你考虑的方方面面，来帮助你找到自己需要的。但是你找到自己需求的前提是什么？你得知道自己需求什么。我也碰到一些公司，说周老师你说我们该怎么做？我说你自己的情况？我们也搞不清楚。你搞不清楚，我怎么知道你要怎么做。所以首先你得清楚自己的情况，然后你还要清楚现在的技术提供一什么。

所以这些基础可能不一定就是在不同环境下可能就不一样，甚至观点不一样，我也碰到了好多矛盾。我的书上有人问我，说你的书上有一些东西说法不一样，我说是，现在不同情况，现在软件工程中客观存在就有冲突的观点，有不一样的东西，他是在不同的背景下面，它是适应在不同的边界范围之内，你得理解它的边界条件。所以得承认这是客观事实。所以当我们应用的时候，一个是现在最流行的观点很多东西你不要违反常识，就说这种常识告诉你是最有效的东西，不要盲目追求一些很出名的，要培养成功，积极思考，你自己认识自己的问题，通过理性思考认识极端的方法，全面方法的作用，然后通过一些合作跟交流的模式去解决一些冲突，得到一定的平衡。得到了平衡解决了问题可能说软件工程处理的很多问题是“满意解”，从另一方面“满意解”就等于不“满意解”，为什么？它是个平衡，它遗留了一些满意的东西，那么不满意的东西你就应该想办法再提高去解决。所以应该有发展的观点，不要觉得协商“满意解”就解决一切了。所以还要有一些创新程度。

从总的说起来，对于每一个软件工程师来说，我们面对一个新问题以后，就要了解这些东西，可能它就是一定会有这样的过程，从一些技术是不知道，到知道，知道了以后要真正理解，理解了实践，实践不一定能正确的实践，正确了实践以后可能又能够发现这对你不足的地方里得到一些创新的东西。这整个过程开始你不知道做事的话，你可能一点约束也没有，就是没有规律的无须的工作，等到了你认识一点，就有一些规律，就防止你一些的错误，可以使你的道路走的更正确一些。这样的规律可能只是一些限制，而一个过程就等于给你规定了一条道路，具体作法就更近了。所以这个过程下一步就希望有一些紧迫性的工作，是能够在新的条件下得到新的规律。所以这是一个循环的不断的过程。

通过学习软件工程我对咱们中国老祖宗我还觉得挺有价值，所以我就画了这样中国最古老的太极图，这太极图我觉得很有新意。整个软件工程里面反映了一个人跟机械的关系，就是一个人跟物性的关系。每一个人的一些工作里面是包含着一些机械的可重复的，机械的步骤，虽然你做一些机械的东西，当你把你的工作能够认识到那些东西是可重复的可以机械化的东西出来，假如人是黑的一半，那么人里面核心里面包含了白的东西，逐步的在旋转，开始人的东西慢慢机械的转换成白的，而且把你的主要的工作都覆盖成自动的东西已经转到另外一半了，但是另外一半里面肯定包含机械做的东西，自动的你说不满意的东西，不满意的东西就是需要你你去创意的东西。所以白自动的里面还包含黑的，不能够一片雪白的，还有黑的就需要人解决的，而人解决了以后对新的问题就循环性的提高。所以人跟机器是人力跟自动的过程是在不同的层次上逐步的演化。

更进一步，很多人都觉得统一跟不统一的东西，咱们老祖宗也解释很好，统一里面包含不统一，它总有一些不太满意的东西，所以统一了以后让不统一的那些东西它会随着环境条件的变化不统一越来越多，就是说各种各样的方法、平台、机器，越来越多，但是不统一的过程中间有它的共性的部分，这个共性的部分如果能够慢慢超越了以后，又会变化统一，所以这又是一个循环的过程。

这是人跟机器的太极图，这是技术方法、体系结构、平台、软硬件划分、硬件、体系结构、计算原则，也是符合这样的规律，比方说我举个例子，太极图大家都知道，这是体系结构，原来统一用什么统一，都是用结构图，四个体系结构。七十年代大家日子过的很好，这个体系结构都是统一的人都画结构图，但是结构图里面都包含很多问题，但是当时不突出，到了现在开始不行了，这个结构图不好了，就要层次结构的，然后“Client/Server”结构的，还有白板等等，就有各种各样体系结构。所以都不怎么样，就是说有很多很多了。这体系结构到了什么时候有一个层次，比方说对突然到了某一个程度以后，什么时候因特网刚起来的时候，什么 B/S 模式，大家都觉得 B/S 模式好，大家一窝蜂都 B/S 模式，这就统一过来了，过了几年，对 B/S 模式又不满足了，又过来了，到了一阵子可能又会统一了。所以统一不统一也是一个循环，所以不要认为今天统一就是一辈子统一，今天不统一一辈子不统一。更进一步可以画这样一个图，也就是说白的里面实际上白的里面还包括黑的，所以外面看成软件工程方法，里面看成公司的软件平台，或者硬件，硬件也再变，各式各样变。所以我觉得老祖宗的东西比现在其他的東西还能够有生命力，这里面能加进西方的女巫通过水晶球看过去跟未来，我们中国人就通过八卦图看过去跟未来。软件工程我觉得跟这个也很有道理。这是我对现在软件工程的看法，最根本的就是人跟物的关系，就是第一个人跟机器的关系，就讲到这儿。

主持人：

非常感谢周老师的精彩见解，从深层次上从思想给我们大家一个启发和提示。

提问：

刚才周老师等于把中国古代的辩证唯物主义用到了软件工程上了。那能不能把马克思的辩证唯物主义也用到软件工程上了。因为您刚才把中国古代的那些很明显的那种，比如说不断发展，不断平衡等等这些理论都已经用到了，我只想知道一下比如说更加现代的一些理论。

周之英：

我觉得在中国古代也包含了这些东西，矛盾、实践这里面都有的。

提问：

您刚才用的是阴阳鱼，就是说不断地旋转变化，马克思主义讲的是宇宙永远的统一和平衡，这样是不是相对来说也是一种辩证。

周之英：

我觉得到目前为止还没有这么来看，是不是转到最后是怎么得我搞不清楚。

我想从目前我们现在能看到的是在变化发展、平衡，这些东西都需要，而且我就觉得实际上平衡中间包含不平衡，不平衡中间也孕育着下一次平衡的可能性。

王东临：

周老师，教科书里面什么时候能够更新过来？

周之英：

我的三本书明年上半年可能有第二版，我现在花比较大的力量更新这个内容，因为这本书出得比较仓促，内容很多，我觉得也不能砍了，实际上更丰富的内容在这样规模下还需要拿出来，第二个，里面有很多问题或者什么也来不及精雕细刻，所以我很欢迎大家如果发现里面有什么问题跟我联系，告诉我。因为微软版本都发给大家，我发给了大家以后，你们假如发现问题就应该告诉我。我现在花比较多的时间是在做第二版的修改，框架还是一样，但会加丰富一些更实际的东西，更结合更新发展的东西。

提问：

刚才您提到很多公司可能不太清楚自己需要什么，这可能是很多目前国内公司的现状，周老师有什么方法解决这个问题呢？

周之英：

实际也是分层次的，不是说我有方法。我觉得所有的管理学什么东西都是有这样思考的模式。比方说公司首先得要确定你商务目标是什么，你要做什么事情，这事情就不是我说回答的，但是你办公司的人自己要明白，你得要确定跟市场对路的产品，或者是服务，或者是产品，你都要确定。然后你怎么样做，就是你企业目标确定了以后你就怎么来实施这些东西，然后你看实施中间有什么问题，然后针对问题去解决。所以现在软件工程改进的话，我最近也看见了一些美国包括 CMM 的模式，它现在也是受到了各方面的影响，它反复地强调，它要是你首先是从你目前的状态做起，然后来分析你的问题，拿它的作为对比来分析，现在有很多方法可以做。但是你首先自己总要做，你什么都不做，然后说你告诉我，那是不可能的。比方说已经成立了公司，他肯定要知道做什么，做了什么东西以后，你有没有提高的欲望，有没有改进欲望，你觉得很好，那你就继续好吧，也不用看别人了你就很高兴了。如果你觉得这地方不够那地方不够，你就针对自己的问题，再有一个你把自己的问题排排次序，逐步地进行解决，这都是软件工程里面最基本的思想。

提问：

现在一般的不是软件企业有很多所谓的管理咨询公司，专门帮助他们发现问题，解决问题，你觉得以后是不是存在专门的咨询软件公司，我帮你发现，我帮你解决？

周之英：

我想是会有，但这个东西对中国会有一个过程，会有一条路，会有一段时间，对这些公司会非常有效。因为我们国家就是在这方面真正的研究工作做的非常少，可以说没有。怎么在中国环境下能够达到这些东西，可能研究地比较少，但是国外有很多可以借鉴的东西，但是我们要正确的借鉴，你别搞歪门邪道，那也是没用的，比方说学 CMM，它是强调给你作为改进的一种测量，测量你改进到什么，并不是要求你追求这多少级的指标，如果你为了指标，你就会弄虚作假，最后你什么也得不到，白花时间和精力，第二对到底有用没有用，所以这些东西都会有国外的经验，但是国外的经验你要消化，包括 CMM 它就是强调，最近它有一个叫做 CMMI 就是集成的 CMM，特别强调你要理解你本身的环境，你不是它的 CMM，不是说照搬条

文，照搬条文必然走到死胡同，而是要理解你自己的工作，建立在这个基础上。所以在这个时候，你比方说有一个咨询公司到你这里帮助你，是可以的，但是你得要提供，你要配合，你要跟他一起商量，但是咱们国家会有一段时间，大家重视了以后会有一批人从事这样的工作，跟企业结合起来做这方面的工作，我觉得还是很需要的。

提问：

你觉得软件工程会不会限制一些软件公司的创新？

周之英：

规则是帮助你，但是你要正确地认识你，就像你拿的文档是给你交流给你总结经验，如果你把文档变成死的，变成压在你头上的紧箍咒，那可能就限制你创新，你要正确理解。因为对于每一个软件公司说规则这些东西不是外加的，是你自己建立的，你不要建立了规则去说服自己，建立规则要帮助你，如果你发现这个规则不合适了，影响创新了，或者你创新需要新规则，你就得要调整规则，所以你这一个规则，这样软件工程这本身是什么呢？是处在发展中间，不是一个死的。也有人来问我，说这东西影响了我们，文档没用，我说你既然你说文档没有用，直说你现在做的事情是没有意义的，你就别做，等你认识到你能做出有用的东西你再做，你去做有用的东西，你干吗做没用的东西，所以这是一种常识，你不要把违反常识的东西。

主持人：

我们再次感谢周老师的精彩见解。第二位发言的嘉宾是微软研发中心服务系统平台的杨永生经理。微软是世界上最大的微软公司，但是微软产品开发流程，还有内部规范一直是很多软件开发者都想知道的，这就是所谓的微软的秘密了。微软内部一定有一套自己的方法，但是他没有公开出来，告诉你第几级应该是这样应该是那样，没有告诉大家。假如微软这套东西虽然说没有像CMM，但是它可能不是CMM胜似CMM。今天我们请到了杨永生先生给大家来展示一下微软的秘密，杨先生目前在微软负责的MSN相关产品的开发。他今天发言的主题是《微软产品开发过程》。有请杨先生。

杨永生：

大家好，很高兴今天有这样的机会能跟大家交流一下。

微软公司一直希望帮助祖国发展软件企业，帮助软件产业，包括鲍默尔、比尔盖茨也在不同场合提到过帮助中国发展软件产业。我们研发中心更的更前一步，在今年7月份的时候，我们和信息产业部签订了为期两年的合同，就是希望帮助国内一些软件企业做一些培训团，向他们介绍一些微软怎么开发软件。目前我们已经办了两期，第一期在北京办的，第二期在上海。我不知道在座的有没有参加过的。作为我自己我参加了一些讲解，所以我也很高兴今天有这个机会，和大家交流一下，看看大家关心什么问题，以后继续再办这样的活动更加有的放矢。

今天的题目写的是微软产品的开发过程，这个题目很大，我们跟信息产业部办的总共是三到四天，不可能在这么短时间全讲完，我主要从开发实践，从过程方面做一些介绍。

我看到王东临发的电子邮件里面提到了软件产业，或者作坊式生产。什么作坊式生产或者说什么时候我们成为软件产业了，我们个人观点作为工业化首先分工特别细化，就是专业化更细化，所有事情向专家方式发展，另外生产过程，质量控制，标准化、产业化这两个标志着一个产业的形成，当然还有一些市场，或者你产业的规模，以及有没有这个行业。

刚才周老师说，软件产业像一些工业或者工业生产，最早的一些领域，但是我觉得更像一个电影工业，我更决定非常像拍电影，首先都是你头脑的产物，然后全是由人完成，需要复杂的过程，电影它本身分工又非常细，有导演，各种制片什么的，投入产出比很大，但是高风险，有可能做一部电影或者开发一个软件，有可能很成功，但也有可能用，但有没有也没有人用，但共同的都有盗版，电影有一些影评人，软件也有什么专家，还是很相似的。看一下中国电影，给我们启发是我们有很好的演员，上次我去美国看到一个，他最喜欢的演员是章子怡，演员是很好的，但是为什么我们电影没有好莱坞，到国际上，我觉得我们角色的分布不够平衡。就像软件一样，有一些我们有很好的程序员，或者其他方面有很好的东西，但是整体上不平衡的，产品工业化的或者推向市场方面，或者产品质量方面还是不够的，另外我觉得生产方式和生产过程不够标准化。

我主要介绍微软内部产品组角色划分，他们是怎么划分的，他们具有做什么，这些人是怎么组织起来的参与产品的开发，还有一部分我想介绍一下微软产品过程，我们有了各种角色，有了团队，怎么把一个项目从始至终把它完成，推向市场。

有一些谈不上微软独有的，就是行业流行的，但有一些与微软企业文化会有一些相关。以前我在上海讲课，他们说你这个东西我们拿回去我们也用不着，首先我觉得最基本一点，我们谈的微软怎么做最终产品，对于做解决方案微软有一个东西，叫“Microsoft Solution Framework”就是解决方案框架。还有一个，微软那么大，你们可以这样做，我们一共几十个人，我们是不是没法做这个事情。这个也不见得，微软组织结构你会发现，虽然是很大的公司，但是实际上，在某种程度上都是以好多小公司方式在运作，具体到一个团队人员规模也并不是很大的。

微软产品组现在基本上有十一类人，各有各的专长，在产品开发过程中有自己各自的不同的任务。

第一个产品规划，产品规划人员主要任务调查市场调查你的竞争对手，客户，以及所有市场需求，是定义产品的过程。他们通过做很多研究，通过跟踪市场用户，做一些市场调查，看一些行业的报告，从而使产品确定三到五年的发展规划，其实作为产品规划人员最重要的一点就是前瞻性，能看到不仅仅是现在市场是什么样的，比如三到五年以后是什么样的，我们可以看到微软好多产品，比如“普瑞申”(?) 3.0，可能 1.0、2.0 不是很好的东西，有可能是质量的东西，有可能是超前的市场，像 Windows，做出来时候市场无论从硬件或者软件应用程序没有这个市场，但是到 3.0 已经是占领了很强的市场，取得了很大的成功，这一点产品规划人员是非常重要的。

第二类人是产品管理，某种程度上有点类似于做传统市场，但是也不尽然，他们主要任务是把产品推向市场，包括产品的定位、包装、最重要一点你像用户发布一个什么信息，用户为什么买你的产品，或者升级你的产品。很多人说微软的产品质量方面，就是市场做的很好，因为我做过 IE，实际上 IE 就是一个很好的例子。在 IE 推向市场过程中实际是很好的成功的案例，比如 IE 最初的用户定位第一用户不是试图让 Internet 已有的用户转到 IE 上，这是从来没有这么做过，而是面向新的 Internet 用户。就是用户定位很清楚。所以从来没有说你是 Netscape 用户，我们比 Netscape 怎么好，而是向简单易用方面发展，像新的用户怎么很快掌握。还有使用“slot”这种可能就是分发渠道，对 IE 每个版本侧重点不一样，就是一条简单的信息就会通知用户，这个版本绝对比其他版本有什么好处。这是产品管理人员的。

下一个角色是程序管理，我们以前谈到说项目管理，但是上次我讲的时候他们说国内项目经理实际上是权很大的，在微软程序管理主要是做产品，最主要的是他的任务是适当的时候推出适当的产品，他最主要碰到的困难就是做出控制，适当时候意味着你必须定好的日程发布，不能

有延误，大家知道产品过程中不确定的就是人为因素，这个发布日期控制好，这是最重要的。还有做出取舍，不可能你永远把所有事情都在这个版本，有些时候你会在发布日期和新的特性之间需要做出取舍，或者采取新的技术，用新的技术，算法什么是不是需要，这些做出取舍，甚至有个“Bug-free”，我们是不是需要去做。因为你需要衡量做这件事情的危险性，需要衡量特别清楚。

这三类人员整个把产品的策划，推向市场，以及产品开发过程控制基本上定下来了，可以说是说最关键的。

剩下的像产品设计，主要是做产品的用户界面或者可视化方面的设计，这些人一般人都有设计方面的背景，特别像微软方面的产品，以前对用户界面或者用户交互方面侧重不多，因为传统PC早期是专业人员的工具，现在越来越向消费者初用者方向发展，那么对于界面设计要求越来越投入了很大的人力。我不知道大家都看到新的Windows XP，或者像“MS-Installer”，这跟传统的产品，外观，包括用户使用是完全不一样的。

产品设计还有一个重要的工作，就是控制你的产品所有可视部分保持一致，像不同的模块或者不同的特性之间有不同的组合，如何保证看起来一样，或者用户不至于在一个产品使用时突然觉得不是一个公司的产品，这就取决于产品设计人员。

第五种人产品可运行评估工程师，这类人主要做的保证这个产品可用，易用，而且能够被用户接受。传统一般在产品开发的过程中或者初期，有一些不同的原形，就是对一些产品怎么做，用户怎么交货，设计一些不同的原形，做一些测试，一般的像单面精审，后面看用户按了哪些键才使用到你的功能，或者你跟你的竞争对手产品比一下，可能用人家产品很快就做完了某种事，但用你的产品就差一些，这些方面就需要你反映到产品上进行提高。这方面微软一直是重视的。你可能注意到在IE早期版本，地址栏里面并没有“Go Button”，只是一个地址栏，但是后来发现一些用户就知道把地址敲进去就在那儿等着，也不按回车，确实就有这样的人，通过运行测试，所以后来在5.0开始在后面加一个按钮，敲完以后用户按一下按钮连到他所需要的网页。

下一类就是开发人员。开发人员在微软应该是很重要的，但是我感觉没有像我们国内一些企业更重要一些。开发人员主要一部分是设计一些算法，对PM做出一些文档或者特性说明要提出自己的反馈，你希望我做的一些事情我是不是能用，还有更重要的一块，就是帮助PM推出产品日程，我们从什么时候可以做到“beta”1、2，这跟实际上开发人员密切相关，开发人员决定它的进度。

下一部是测试，微软对测试非常重视。我们在人员结构基本上会“迪外勒特”(?)是一比一比的结构。测试人员在产品开发过程中要独立完成，就是不受不管是TM他们的影响，独立完成测试。另外某些情况作为用户的代言人，也就是用户的利益，如果你认为这个产品这样发出去不行，往往就引起有一些激烈的争吵，决定缺陷到底是要不要。

再下一类就是微软特有的就是本地化。我们中国微软研发中心也是有一些人本地化，这一点我想对大家不是很适用，我们将来怎么把我们产品推向全球，有一个全球化的过程，微软为什么重视本地化？微软现在60%的营业额来自美国以外的市场，再有一个文档发布，这里面包括网站方面的文档，这些文档主要是帮助用户怎么使用产品，还有面向开发人员的，做一些例子，这是主要的。我们传统谈到的开发文档在微软是PM来完成的，就是说所有的程序经理在开始针对每个特性写非常详尽的文件。

还是产品支持，就是售后支持，这在微软也是非常重要的，一方面微软跟最终用户打交道最常

见的一个途径，往往有很多用户打电话提出这个问题，将来在下一个版本会把它解决掉，还有最重要的在微软来说每打进一个电话都是要花钱的，实际上产品支持直接影响到公司的营业额，怎么提供更快速更有效的用户支持，这是最重要的一块。

最后是运营管理，就是网站运营管理。大家也知道，微软产品目前越来越多和因特网集成更紧密，比如我现在做的一些“Hotmail”或者“Microsoft.com”这些本身就是一个网站产品，像运营管理角色原来是没有的，这只是近两三年来新发展出来的角色，这在将来会越来越重要，因为你跟传统的所谓我们做CD包装的产品不一样，你在做连接在线服务时，传统的你说我把CD烧完了，发布了，那已经没有时间，那用户买了已经赚钱了，实际在连接服务的时候，在你软件发布时仅仅是一个开始，用户只要使用一天你都需要花钱，都会影响你整个的盈利。实际上在线管理是非常复杂的，比如“Hotmail”，“Hotmail”在前端大概可能有五百多个服务器，因为“Hotmail”现在有一亿一千万用户，由五百多个服务器来满足用户，就是你登录上去看到这一页有这些，现在全世界可能有两千个服务器，后台还有无数的。怎么把这复杂的系统，实际上很复杂，因为有底层的网络，有硬件，还有操作系统，还有上面的你的应用程序，加上Internet本身是不确定的环节，怎么把这个环节管理好，是很具有挑战性的。因为用户跟传统的应用程序是不一样的，用户随时可以走开。而且还有很多不确定性，包括我们可能传统的卖CD，那么我卖的越多我赚的钱越多。但是在连接的时候，用户多有时候也可能是一个问题，就是你可能支持不了那么多用户，比如像9.11，有很多人来访问，你是不是能满足这么多用户访问的需求。网站运行往往需要提前对流量，或者对用户数满意做出比较精确的预测。

我记得在去年Internet比较热的时候，当时在硅谷时你如果把你的容量要扩，你要提前6个月提出申请，否则就根本没有那么多可用贷款，现在可能好一些。在运营管理在微软也会越来越重要，你产品的很多设计会影响到你到底能不能好好运行。

所以现在基本上运行管理、产品规划、产品管理和程序管理，这四类人实际上在主要来推动产品的进程，其他人是一个被动的或者具体的人，但整个推动是这四类人来完成的。

前面我们讲了微软现在基本上有十一个工种，怎么把这些人组织起来，能够更有效地去投入到开发过程中。微软目前基本上是一种所谓的条和块的结构，在公司内部最基本的组织是一个产品单元，比如像IE，IE是一个产品单元组，产品单元组的管理者会有预算，他有人有钱，也会负责一些究竟产品能不能赚钱，负责营业额。在每个产品单元内在行政上按你的工作类型，像项目经理他会有一个最上面有一个总的，底下有一个小经理，比如开发人员有一个开发组长，或者底下跟测试人员，这是在行政上，行政组织结构主要是为了对你的考核做出一些考核，包括将来会不会给你加工资。在做产品的时候，在每个产品单元内，按不同的特性划分各个不同的项目组，划分的基本原则希望由一个很精干很小型的团队来进行开发。因为我说了按产品的不同特性来划分，这样要求你在产品设计时，大的产品能分成小的模块和小的特性，然后相互之间又没有很大的依从关系，就是你跨组的交货或者跨组的依从关系是你最难管理的，往往你请别的组人做什么，或者希望他帮助你什么是很困难的，因为大家都有自己的事情。每一个团队内基本上项目经理，或者程序经理他来领导一个特性，他来负责特性，下面会有开发人员，也会有测试人员，基本上开发人员和测试人员的比例一般都是一比一，这一组的概念差不多十个人，是最基本的开发单元。做一些跟技术有关的决定基本上是项目经理做出来的，不会有上面的人左右你的决定。这种组织结构能够使在一些生物和技术方面很快做一些决定，因为人少，还有能使大的团队能像小的团队这样很快向前移动，效率不会受到影响。

举一下IE产品组，它在不同时期有不同的人员，人数是不同的，最早IE 1.0是买来的产品，几个人，IE 2.0可能是三四十个人，到IE 4的时候基本上就到了300人的项目组。在300人的项目组里面是这样的组成，一个是产品单元经理，这因为是以产品单元为最基本单位，所以产品单元经理是大老板，下面有五个产品规划人员，产品经理可能有二十个，

项目经理五十个，开发人员一百个，测试人员也有一百个，还有文档发布，因为 IE 也有一些 SDK，或者也有一些连接的网页，还有自己的帮助，文档人员有十个人。这种组织也是根据产品的特性，或者你在这个版本中间你的侧重点来决定的。同样在 IE 5.5 的时候，IE 5.5 也有 300 个人，这时候项目经理就只有 15 个人，比 IE 4 五十个人要少好多，开发人员也只有 40 个人，因为到 IE 5.5 的时候，基本上大的特性基本上稳定的，IE 5.5 希望面向最终用户方面做的工作很少，主要在稳定性和性能方面做一些，另外对一些公司大企业的用户做一些支持，所以开发人员和项目经理数目减少了，但是测试人员很多，测试人员有 200 人，这主要是在 IE 4 的时候觉得少，所以在 IE 5 的时候就组织独立的测试队伍进行测试。

IE 产业组分为十个项目组，每个组大概有十到五十个人，基本上负责一个产品模块，像流浏览，或者 HTML 的编辑、打印，这基本上就叫做一个项目，每个模块可能会有十个人左右。基本上是这样。但是有一些时间一个项目经理会负责不止一个特性，甚至有一些开发人员可能他在某些方面有专长，他也需要在不同组织之间流动，基本是一个动态的。

微软产品开发过程中的策略。开发过程基本的策略或者基本的原则是希望把大的项目分为若干个里程碑似的开发周期，并在各个周期都要考虑一些冗余，因为给定的日期永远不够，所以都要考虑一些冗余，就使你的开发周期变的更实际一些。内部有一个目标描述实际上就是我们这个产品是做什么，一般都是很简单一句话，这一句话来指导你产品整个开发过程，包括你加哪些特性，所加的特性要和你产品达到的目标要相关，如果没有任何关系，或者关系不是很大，那么可能在砍的时候先把你这个特性就砍掉了。产品目标描述是保证所有的不管你是多少个人的组织是沿着同一个方向发展。

还有产品特性，这可能是我们传统意义上的产品特性描述，或者传统意义上的文档，某一个特性我们决定怎么做，里面技术规范写的非常详细，甚至详细到可能每一个对话框框从那儿到那儿是几个像素，或者有哪些没有解决的问题，包括修改的历史，什么时候做了一次修改，改了哪些，这个微软一般叫“斯拜沃”（？），或者叫“斯拜沃斯 K 申”（？），这东西是指导开发过程中所有的人都会参照它，一般是由程序经理或者 PM 来书写，但是像“迪外勒喷”你会在写的时候你会参照看一看，你怎么做，包括 testing，假如发现一个问题，决定应该是这样子还是一个缺陷，往往都会以“斯拜沃”（？）为参照，这个实际上最终最权威的技术，所有人看同样一个东西，虽然大家分在不同组，但是大家做的方向是一样的。

最重要利用用户的数据，由产品规划人员或者市场人员，或者技术支持人员反馈的数据来决定一些特性的取舍，或者优先级的排定，我们加不加这个特性，不是开发人员觉得好，我做这个东西，我在技术上能达到什么，往往还是从用户角度来考虑，用户从中间有多大收益来决定。

还有更重要一点就是统一的术语，包括在微软内部刚进去时也会做类似这样的，他会请的各种决策做一个讲座，大概需要六七个小时，其中有很多术语，就是公司自己的企业文化，有很多缩写或者什么的，还有对这套开发模式必须保证所有人理解的都是统一的，这样无论你在做事或者谈起来的时候，你才能保证大家理解是一样的。

还有一个在开发产品过程中不间断地测试，并不是做完了到某一个阶段才开始测试，往往那个时候已经晚了。

微软开发过程分为四个阶段，第一个阶段是规划阶段，这个阶段基本上是由产品规划人员以及项目经理来驱动的，这个阶段主要是要完成这样一些事情，一个是目标描述或者，产品目标，比如竞争对手一些信息，我们产品要达到什么目标。第二部分基于这个产品目标我们已经知道了，我们需要做哪些事，做哪些特性来达到这个目标，这决定了产品提供哪些的功能，然后 PM 就要根据这个功能来写出相应的规范。一般产品规格说明也可能就是传统上说得技术文档，

基本会两次写，第一次写一个简单的一页的，里面就列出了你这个功能或者你的特性希望达到什么要求，跟我们整个产品的目标有哪些相关的，或者产品之间依从性，为什么要做这个，做完之后会列出很长的单子，基本上每个人希望做的事情越多越好，或者对一些项目经理，如果你没有特性去负责，你可能就没有工作了，你的特性被砍掉了，往往意味着一些人就许多去做别的事情。所以一个人人的天性做更多事情，然后从关系到切身利益，往往有很多的特性，但是写完这一页之后，大家会坐在一起看一看，排定一个优先级别，哪些我们先做，哪些有可能做，或者哪些是下一版本，把这个事情做完了，程序经理会写一个更详尽的，这个指导开发测试整个过程的技术文档，基本上一般都有模板，像 I E 可能特殊一些，安全性会考虑一些，或者产品之间依从性考虑一些，甚至结构问题，各个部件的相关性这些都会谈到。

在规划阶段，所有的特性规格说明完了以后制定日程进度表，我们什么时候把这个产品做完，需要多长时间。这个日程进度表往往需要由开发人员的参与，你看到了这些产品规范，根据你的经验估计做这个需要多长时间，还需要打入一些冗余，把这个做完之后，产品规划阶段就已经完成了。

产品阶段完成的标志，就是所有特性说明以及“所维申斯迪特门特”（？），以及大致的日程完成了标志着进入第二个阶段就开发阶段。因为我们自己有特性，已经知道做什么，根据这些特性会把它分为三到四个阶段，一般的就“马斯汀 1、马斯汀 2”（？），基本的原则可能重要的或者相互依从的特性一开始做，剩下的一些会在第三个做。这些由开发人员去推动，所有的开发人员开始写代码，开始比如有“Build”，就开始 C H I C K I N，另一方面我刚才谈到有开发人员和测试人员的比例是一比一，对于每一个开发人员都有相应的测试，他写的这些 Code，他把他 C h i c k i n 到一个 build 里面去，或者这个对应的开发人员、测试人员他就会去测一下，就保证你 code 在你子级上是正确的。这个阶段完成的标志是叫所谓的特性完成，或者叫代码完成，所有的这种特性都已经完成了，就是进入了下一个阶段，也就是说测试阶段。测试阶段主要有测试人员，在开发阶段也有测试在进行，在测试阶段进行的主要是集成的，在开发阶段可能各个子模块之间的，在测试阶段是各个子模块做成大的 Build，可以安装，做一些兼容性的测试，性能，或者其他，这都会在测试阶段做，此外发放一些“beta”，或者公共的或者范围小的“beta”，放在网上让用户去做，这一阶段会有更多的“bug”进来，但是这一阶段基本上不会增加一些新的特性，因为已经太危险了，就是到了稳定化阶段不会加入任何新的特性。这一阶段的标志是所谓的零缺陷，因为微软有一些来跟踪缺陷或者“bug”的工具，如果这些工具看到针对这个发布周期已经没有任何活动的 bug，这就标志着稳定化阶段已经结束。

现在一个趋势，就是稳定化阶段做的越来越长，这是一个趋势。到了零缺陷这一步就进入了下一步的发布阶段。现在你到了零切线，但是过两天还有问题出现，在这一阶段大家会看看这是不是会影响到我们的发布，有些可能会推到下一个版本，重要的会再做一个 build，一直确认这可以发布了，才发布。传统的一般做一个 C D，或者把它发到网上。

基本上有四个阶段，规划阶段，开发阶段，测试和稳定化阶段，最后发布阶段，最后发布阶段会有像产品经理、项目经理，以及做运营管理的人由他们来执行。

总的原则在微软一个是有详细的分工和职责的划分，通过各个人人的角色控制产品开发过程。我刚才谈的四个过程，十一个角色，但是每个角色实际上由不同人来推动，比如像产品规划人员，其实在第一个阶段和第二个阶段产品规划人员会有一些工作，到第三个阶段因为特性已经完成了，不会有新的特性，产品规划人员已经做下个版本，但是产品经理会继续把这个产品保证这个产品继续进行。还有客户需求决定产品的方向和目标，往往做一些规定或者做一些取舍考虑的是客户和市场，很少纯粹为了技术和其他原因。最重要的是把大的项目分成若干个子项目，是渐进的，不是一下子把很大的问题解决。还有目标描述和产品特性说明，就是我们传统

文档，这是为整个项目起到了指导作用，这必须定的很清晰，让所有人都会看到它。最后一点，从项目开始让所有人都去介入，你可能会看到有一些人介入地不明显，但是基本会把他参与进来，因为有一句话说好的产品是设计出来的，不是最后开发出来的，因为前期基本上定下来以后，开发的后期是完成的阶段，如果设计有缺陷，比如没有考虑到技术支持方面的问题，后期很难做，假使再加进去对产品质量或者发布日期都有很大的影响。还有通过不间断的测试来保证产品的质量。

我就讲这些，谢谢大家。

提问：

怎么看待 I E 的“帮助”中的“N e t s c a p e 用户”项的设置？

杨永生：

我没有说让“N e t s c a p e 用户”怎么切换到 I E 什么，他看到在工程在 I E 里面有这样一个设置。我谈的是在公开的，比如说我们做“浪驰”，或者做一些广告从来没有针对“N e t s c a p e”这样简单的小的用户群体，而是针对更广大的用户，当然里面加了这样的设置，只是方便，并没有特别的，你可能看到跟“外沃卡姆拉德”或者跟其他的，或者是 Palm，可能在广告里面会什么，有一些时候比较不客气的，另外我很大程度是市场的问题。我在这方面说的也不是很确切一些。

提问：

微软产品开发过程的每个阶段，包括每个工种，你说有十一个工种，他们之间怎么良好沟通？第二个问题，微软这个过程有缺陷吗？

杨永生：

第一怎么保证他们良好的沟通，在这个过程中，项目经理他起到很大的作用，他基本上是交流通信的中心，来做一些决定，他负责跟各个地方的沟通，主要是 P M 做的，他们之间没有特别直接的沟通，有一部分是通过项目经理协调的。P M 是测试或者跟开发。

缺点肯定是有。微软本身并不是一开始就这样的，包括 P M，在 8 9 年做了一个“矛汀普雷德”(?)的项目，在中间发现效果很好，才开始在这儿，包括“E 塞沃”(?)、“E P 瑞斯迈乃之”(?)，以前根本没有这个角色，现在也已经开始了。

缺陷就我来看，我现在确实看不出来。另外这种过程目前来说最适合微软的发展。作为微软基本上做完任何一个项目都会有一个事后分析，大家可能对过程中的一些控制，通讯沟通方面来做一些反省体会，但对大的模式，因为原则就是把大的放到小的，或者各个阶段有专人控制，有不同的侧重点，这个原则大家还是认同的。

提问：

微软为什么没有采取 C M M 方式？

杨永生：

我的感觉是 C M M 只是做一个规范，另外你做项目的时候……反正微软目前不需要用这个来证

明什么，我觉得是这样。

提问：

目前管理或者各大单位都成立了相对独立的软件测试机构，您认为软件测试机构对软件质量会不会起到非常重要的作用？就是目前的大单位可能下属好多独立的小单位，其中并行单位有一个是测试机构。

杨永生：

微软小的时候是全公司只有一个测试队伍，从行政上来说是这样的，后来公司发展大了以后发现这样有问题，这样把全公司一个发展到现在每个产品单元有一个，我不好说，但就国内的，就微软来说这种方式更有助于测试人员和开发人员之间的沟通。

主持人：

下一位嘉宾金山词霸研发部技术总监邹涛先生。金山公司是我们国家最大的通用软件公司，像家用软件，金山词霸，金山影霸，还有游戏软件等等，杀毒软件像金山毒霸，虽然面对世界上很多超级公司竞争，比如微软的竞争，但是金山公司一贯给人印象是豪情万丈。我们现在来听听金山背后是什么样的开发团队。因为今天邹先生忙于开发也没有做好主题发言的准备，他的意思是让大家有问题就问他，有关金山产品方面，比如什么样开发模式，比如对 CMM 怎么看，还有产品可用性这方面，大家都可以提问题。

邹涛：

大家如果对金山研发感兴趣的一些问题可以提一下。

提问：

金山我认为在国内是做得最好的通用公司，现在产品战线已经拉的挺长的，做得可以说做的都算精品，而且按照中国程序员思路都是习惯于单打独斗，金山在这方面组织程序员集合开发相对比较大的软件产品是比较成功的，希望你谈一下这方面的经验。另外一个问题我向请一下微软给金山公司微软给金山公司造成的压力和你们的对策？

邹涛：

就我自己感觉国内微软开发规模相对国外来说基本上还是处于最多是一种中型规模，所以你刚才谈到人的作用，我觉得在金山还是很大程度上依赖于某些技术上比较强项的个人，虽然他也是一个团队在开发，但是个人的作用还是非常非常大的。

对于第二个问题，微软比我们大很多倍，他给我们造成多大的压力呢？因为大家知道金山它现在目前是四条产品线，办公产品、杀毒、娱乐和工具。微软给我们造成的压力是办公，至于压力多大？看看微软和金山大小比较一下估计就是那么大了。

提问：

刚才微软说了一下他的开发秘密，比如有十一个人，比如各式各样的东西，金山有没有这样的一些概念？

邹涛：

我简单介绍一下。作为金山开发的特征性跟微软很像，它主要是产品化的方向，基本上我们从产品做计划，规划，然后软件先写一个计划，编写代码，最后测试，发布维护，流程和人员也是这样的，但是规模和人员没有微软那么大，我们到现在也没有一个开发组有一百个人程序员做一个产品，是没有的，我现在在金山词霸工具产品组，目前也仅仅是三十个人，包括文档，包括程序人员，程序代码。

提问：

您现在单独有一个测试部还是每一个产品组都有一个测试？

邹涛：

我们每一个产品组都有测试。因为这四条线特征不一样，最主要是便于测试员和技术人员之间的沟通。

提问：

国内的企业内能保证测试和开发一比一的比例呢？

回答：

金山公司保证不了。我们产品“红色正版风暴”卖到 2 8，现在 3 0 几，4 0 几，这个利润很低的，所以很难按微软标准维持开发，我觉得在中国如果按微软的开发模式开发软件它肯定要赔钱的。

杨永生：

所以微软可能也在赔钱。

邹涛：

微软可能从全球统一战略出发，所以它就可以容易做到这一点。

周之英：

为什么现在国内软件公司线铺的很开，为什么不集中搞好一样，比如你铺了四个生产线，每一个生产线的力量都不是特别足，这四个生产线差距都非常远的东西，不是差不多的生产线，这样的企业策略你觉得给你们带来更多的机会还是限制了你的发展？

现在软件工程要特别强调针对你的目标。

王东临：

这个问题应该是金山公司董事会来回答这个问题。作为开发部的任务就是贯彻董事会的决议，把董事会要求的决议把产品开发好。

邹涛：

我最多的经验只是在产品开发过程中的开发，谈到更高的策略我就不能回答这个问题。

王东临：

我问一个大家都很关心的问题，咱们软件开发管理体系国内很多都是软件工厂式的，很多人说这是理想状态，但实际操作并不现实，金山公司管理模式是什么样的模式，能表述一下吗？在产品开发的管理体系。

邹涛：

按现在来看，我们公司开发模式更有点像CMM的模式，而且我们目前公司正在通过CMM认证，我们这样做的目的不是说我们准备做项目或者做什么，像刚才有个嘉宾问我，说能不能出现一种咨询公司帮软件公司提供整体解决方案之类的，我们出CMM目的也是规范一下软件开发者的管理水平。所以基本上目前也是套用这套管理模式。我们现在从CMM2开始做。可重复性的，因为它是成熟模式。

提问：

能不能具体介绍一下你们公司在组织和人员划分方面的情况。

邹涛：

目前有产品经理、项目经理、项目开发人员、测试经理、文档，还有最基本的五个模块，要细分还有很多角色。

提问：

我想问一个企业文化方面的问题。软件工程里面涉及到管理和人的问题，金山公司在国内软件业做得是最好的，管理和人还有激情，因为做软件过程中，激情是很重要的东西，你们在管理过程中如何协调管理、人和激情这三者之间的关系呢？

邹涛：

这个问题我只能从我自身来反应一下，可以说在中国恶劣的软件环境下，会来从事软件开发的一定是对软件充满热爱的，很有激情的，所以任何一个加入软件公司的年轻人当他刚刚走进这个大门时，他肯定充满激情的，但这种激情能维持到多久，这就涉及到对人员的管理。我们公司按我们雷总说金山是一个舞台，你在这个舞台上你唱戏，能唱多大就唱多大。这就是金山给员工一个非常好的前景，只要你愿意去做，只要做好，你就有前途的，我们的管理就是基于这个思想，紧紧围绕这个思想。

杨永生：

微软招人的时候会看看你是不是想做这个事情，还是仅仅因为微软才来这个东西，另外靠一套完整的机制，包括奖惩激励机制，以及使每个人感到被尊敬，如果他有什么想法一个可以加入产品中，或者可以去高层反映上去，主要是指这两方面。

王东临：

我理解的是，软件开发在很大程度上还是有一定创造性的劳动，如何在规范化可重复性的过程中，如何避免赏识创造性的压制，我理解是这个指这个意思。

杨永生：

你觉得在开发过程中，如果你仅仅是这个，因为在微软如果做到特性组，你完全可以做一些自己决定，你完全按照自己的做法做产品，让你做什么已经定下来的，你自己在技术方面只要好的话你完全可以。

王东临：

比如说做文学创作，比如像文章这样，批准咱们几个人多长时间写出几十集电视剧，那么工厂式的那种，批示你生产电视剧，他们就很难写出像《红楼梦》这样优秀的文学作品来。同样像很规范化的软件开发管理过程中，如何保证它的质量，也能达到很高的创造性产品出来？

杨永生：

第一给他做好的机会，另外微软内部有很多你可以内部轮换，你有很多工作机会，如果你做的好你可以做别的事情，如果你真的有想法，并且得到别人的认同，你也可以开展项目，这种事情是有的。像项目 X b o x，原来实际上是几个工程师闲暇时间做的东西，只要他能说服上面说这个东西确实有市场有生命力，你就可以把这个想法更你一些东西。

提问：

刚才微软那位先生说，软件开发过程跟电影行业这是一个挺好的比喻，因为都涉及到人，但是人的能动性都在里面，但是软件开发过程中可能跟拍电影有些类似，有一些时候我们所谓的软件高手，好比我追求一种共性，追求一种工程化的东西，就要抹煞一些个性，因为每个人都有鲜明的个性就很难有共性了。怎么在开发过程中，特别像金山公司微软公司，在开放过程当中怎么把程序员，特别国内的程序员一般自视清高的现象特别严重，怎么把个性和公司整体，包括项目产品追求的共同利益结合起来，因为有的情况在产品过程中追求最大的共性，追求产品的特性，而必须要牺牲一部分人自认为非常好的想法，这是一种矛盾，而这种矛盾多了对公司本身也不是很好的事情，那么二者是怎么有机地调和的？

邹涛：

这种问题我们也经常碰到。你开始讲了大家都是很有个性的人走到一起来，事实上最后大家要慢慢的有一部分的共性，因为大家是在一起做事情，如果大家我说这样做，你说那样做，这样很难在一起合作，很难做出产品来。所以我们对程序员要求都有必须规范的要求，当然你可以有自己的想法，想法是想法，但是在编写的规范，或者其它流程方面你一定得遵守，你的想法可以用规范表达出来并不会规范规则，因为你产品涉及到不是说做完就扔在那儿不管的，它需要后期的维护或下一代版本的，所以它的可读性和可维护性是非常重要的问题。我们公司处理基本就是这样的，还是需要你个性稍微收敛一点点，融到共性当中去。如果你的想法特别新颖独特.....

我还是没有搞清楚他的想法还是行为？（回答：行为）行为又表现在他编程的行为还是为人个性的.....这个不好说。编程行为肯定是协调一致的，你的想法实现的方法很奇妙，当然大家就用你这一套，这是不矛盾的。如果你这个人个性比较古怪，很难和人合作，这个我们基本上不会把他放到团队当中来。

提问：

项目经理他写出需求文档，他定出详细技术规范，这个技术规范会详细到什么程度？

邹涛：

我得说明一点，刚才杨先生说的我理解，我们这个公司是什么，项目经理写得并不是编程规范，编程规范是程序员日常编程的准则，项目经理写得是针对产品来说，根据用户需求分析，然后设计出功能点定的规范。假如你有一个窗口，点这个纽它会产生什么结果，是这个规范，而不是技术代码怎么写，这两个规范是完全不一样的。

杨永生：

产品说明规范比这个更详细一些，最简单的可能对话框什么样，最复杂的有些可能会涉及到可能用哪个函数去实现可能更合适一些，前提是把这个问题的描述特别清楚，开发人员看到了以后，可能会去设计算法，或者能去做，而不会反过来再问你。

提问：

说到算法，软件开发过程中，软件有两个大类，一种要求是技术，技术含量特别高，第二技术难度不是很大，但是需求特别大，软件工程概念是不是特别适合技术难度不是特别高，但是需求特别多特别复杂的需求。如果涉及到一个算法，像微软做的多媒体方面，比如这方面的编写，这种算法和这个过程来说是不是用到软件工程的观念？

周之英：

我想软件工程是在对所有的软件各种情况都是有用的，不光是对特别简单的，但是你在一些比较复杂问题的时候，可能你不是那么容易得到一个机械的步骤直接就到底了，软件工程东西主要是帮助你解决一些无效的工作，就是提高你的效率，并不是替代你的工作，就是把一些混乱的东西变成有条理。所以如果你有创意，有些人创意一会儿想到东，一会儿想到西，比如想到17、8遍，最后想出一个解决方案，如果你有软件工程的思维规律，可能你就会缩短这无效的搜索途径，你可能就会用更直接的方法去得到它，减少中间很多不需要不必要的工作，你应该从这方面来理解。不要把软件工程看成一个架在你头上的紧箍咒，这是你自己加的，你何必呢？所以你要用这种思想来理解。

我随便举一个例子，我的学生中间软件工程素养很差，为什么呢？咱们国家对软件工程训练很少，这不光是中国的情况，现在国际上也在讨论，就是大学里面教了学生编程序，但是没编学习怎么样工程化的程度。我有几个学生，我规定他们两礼拜给一次报告，就是他们简单做了哪几件事情。他们报告有问题我马上退回来，他取文件，这个文件愿写什么名字就写什么文字，我就觉得很烦，这一大堆同学，今天“D Z I . c o m”，明天设计 . c o m，我也搞不清楚，我就要求他每一个文件给它一个规定，就是说学生的姓加上你文是什么，是小结就写小结，是设计写设计，是需求写需求，后面是日期，如果你这一天来了两份，第二篇再杠一杠二，这样就清楚了。这就反映软件工程，告诉了这学生，这学生他头一天这么来了，过了一礼拜又来了“D Z I”了。这是什么呢？这不光是我的问题，而是你自己的问题，你自己下面这一次跟那一次到底有什么差别，为什么我把你打发回去了，你自己可以有一个记录。这就是软件工程。软件工程的思维就是把你无效的劳动去掉它，否则我比如一查，可以查到我要的东西。所以种种规定要来帮助你，你说服了我的创造力，我文章非要写“D Z I”，这何必呢？你要这么来看

它以后，软件工程是什么？有很多好的东西来帮助你。至于你是不是需要每一个文档都要规定，首先要写什么名词解释，二次总体目标什么，这倒不一定，看你具体情况，如果有一些情况需要，有一些情况不需要，有很多东西，关键是你掌握它的思想。它是要把你的无效劳动去掉点，你自己总结总结哪些是你的有效劳动，哪些是你的无效劳动，把这些东西弄清楚。

主持人：

下一位主讲嘉宾是摩托罗拉全球电信解决方案部网络方案技术中心的高级经理胡大庆。摩托罗拉网络研发中心已经在2000年通过了CMM5的评估。CMM是目前国际上最实用的软件开发过程标准，和软件企业成熟度评估标准。目前我们国内一共有两家企业通过CMM的评估，两家都在摩托罗拉，另外一家摩托罗拉的中国软件中心。胡大庆先生毕业于清华大学，多年来一直在中国、日本和美国公司从事软件开发和管理工作，网上有一篇被大量转载的文章，文章题目叫做《实施CMM是一条没有终点的路》，胡先生在文章里面发表了许多精辟的见解。他今天发言的主题是《CMM与软件过程改进，需求管理及软件产品，质量保证》。现在有请胡大庆先生。

胡大庆：

大家好。很高兴有这样机会，因为过去虽然都是中国人，都在海淀区，在中关村这一带，但是隔行还是比较大，交流机会很少，今后多增加这方面的交流。因为从任何意义上讲我们也是一个本地的公司，从国家信息产业部的文件规定，像我们公司的性质也应该纳入到国家软件行业管理的规范之内，从人员构成来讲，现在百分之百都是本地人。

我刚才也听了像周老师讲的观点我非常赞成。我们在实施过程中是在99年还是2000年初，也请周老师给我们做过软件工程的课。现在经过不同的渠道，我们在第一线进行实践，周老师在理论上继续深入研究。但是我刚才听了一些观点非常一致，我也感觉有一些安心。

刚才大家的问题好象更多的是在企业管理和运营方面。今天也不是专门来讨论CMM，我只是把软件工程具体实现方法的其中一种有可能选择CMM，这方面稍微谈一谈。

我觉得你是否选择CMM，或者有些企业已经用了ISO，或者其他的标准都是可以的，而且可行的，而且相应收到效果的。对于CMM，我希望大家应该有一个正确的认识，第一个认识应该纠正，我不是说在座，至少是媒体上有很多人有这么一个想法，就是说比较简单把CMM作为一个认证，就像ISO这样，实际上CMM绝对不是一个认证，CMM它有五六个特征叫其他的也好，它其中有一个可以作为认证的标准，但是现在我们过于突出了其中的一个特征了。所以这样如果你没有一个具体的按照CMM的框架，按照它的目标来实现，没有实践积累的情况下，单纯追求第几级是非常危险的事情，有可能导致你正常的失败，因为CMM的引入有可能要求你做很大的改动，任何企业原有的一套是基于很有价值的，如果你把它轻易放弃，引入另外一套东西是非常危险的。所以我觉得CMM更多的要具体的实践，看是否符合你企业的需求，在一段阶段之后，自然而然你可以把它作为一个手段去检验你目前达到的水平。

另外CMM确实不是尽善尽美也不是包罗万象，CMM从出来也不断版本更新，它还是代表了广泛的软件行业的共识。它是通过一些统计的规律，或者如果我们对过程改善作为努力的话，作为我们比较有效的工具。就是说这工具是不是特别适合你，或者说你使用了工具肯定有效，这也没有任何人来保证这样做。

另外还有一点，我觉得CMM核心有一个成熟度的级别，无论你说几级，它之所以定位这样的级别，也是衡量你改善过程中你自己要知道你究竟处在什么位置，现在不应该把级别本身作为

追求目标。应该是把采用 CMM 或者说改善和你企业最终的或者最初的业务目标紧密结合起来。比如说微软为什么不用，从它的发展过程，现在的规模，或者企业的市场的形式决定了他们完全没有必要采用这种，但是不能说，微软没有采用 CMM，CMM 就没有用，这完全看你企业怎么想怎么做。

如果采用 CMM 的话，根据我们的经验，我也简单说一下。其实 CMM 很大的特点可以裁减，就是说它给了一些不同的关键的时间叫 K 级，它同时可以说你企业有一个共同规定好的东西，并不是所有项目都要遵循一套东西，因为不同项目周期规模都不一样，它采用生命周期都不一样，你不可能要求所有公司都要用同一套东西，所以你可以裁减。这东西毕竟不可能是非常完善的，所以大家反过来不能对软件工程，尤其对具体方法有这么苛刻的要求。

另外在实施过程中，大家应该根据企业的需求，从小的范围内，从低的级别来做起，而不应该大张旗鼓的太厉害。因为 CMM 本身的几个级别也是根据多年考虑，工业界的反馈和意见，大家认可的，认为先做哪些，做到了二级，如果完善比较好的，把具体经验再推广到整个企业和各部门中，然后再加一些调整就到了三级，四级就更进一步实践中加上一些更多的定量的数据分析，还有一些东西，可能要求更高一点。

所以它要求的是一些比较小的，在你已经做得很好的基础上，按照 CMM 规划来做小的改善，而不是有一个巨大的变化来取代，这是它的一个特点。

当然它也需要如果参加的话，应该大家都有一个共同目标，要积极参与，同时对所参与的人员都要给予相应充分的培训，对企业来讲，关键要考虑到你的投入，因为软件投入主要是人员，而人员主要投入是时间，如果没有任何投入，你就奢望有一套东西能引进同时促进你企业有一些变化，这是不可能的。

在实施过程中还应该加一些比较高的工具来帮你实现，比如说它其中有一个 SCM，就是“configuration”，这必须依靠工具，因为你不可能靠手工对版本进行管理。

第二个问题，我看到大家有一些观点，把需求的管理和最终推向市场的产品好象割裂开来，甚至处于一种矛盾，就认为需求有什么用，或者需求没有必要做的怎么样，最终满足客户的要求的市场。其实我觉得这本身就不是矛盾，因为就我个人来理解，所谓需求管理就是包含主要的比如说一开始需求的定义，在定义之后在整个过程中，对需求变更的控制是非常重要的。需求定义应该建立在与客户，当然有一些做商业产品不是只对直接用户，就是针对市场，在充分交流基础上来反映出需求，这是它的定义。定义好之后，一系列的设计、技术、文档、编码都是按这个来做。需求的变更完全需要遵循一套已经定义好的过程来做。而不是说非常任意的或者是随意的来进行变更。因为我们最终不需求，把需求文档当成摆设，所以才导致需求有什么用，正因为你对需求变更没有做控制，因为需求应该随时反映出你和客户双方的认可，对需求最新的变化，而这里应该说双方认可，而不是说只要客户要求变化了你就必须变。这个大家要注意，一定双方要进行真正沟通之后，因为有时候客户要求是没有道理的或者是不切实可行的，明知道你做不出来，那么最终结果可以想象。

另外从需求开始一直到最后，我觉得整个过程应该有一个非常严格的叫做可塑性的管理，英文叫（英文），就要一直可以不断地检查，他要求你做的每一件事情都是前一阶段都定义好的，而不是需求，分析是分析，等到设计时我已经走变样了，因为客户来了我需求不管了。那么设计出来的到编码那么过了一段时间又变了，所以这样的话，那确实需求就是一句死的摆设。我想正是因为这样，刚才周老师讲了，虽然咱们确实对一般性可能不太好管，但是对这么具体的东西定义的还是比较清楚，对企业是非常有帮助的。这也是对我需求管理想到的一些。

还有一个观点，我看议题老是把文档和代码对立起来谈，分开谈，至少我不是这么认为，它们都可以统统纳入到软件的产品，软件产品应该是一个比较广泛的概念，它即包括最终交付给用户的，你可以交付源代码，交付其他东西，其实在整个开发过程中，你所相应产生的任何附带的产品或者阶段性的产品，包括你开始的相应的计划，你的需求的分析，技术文档，技术的报道等等这些都是产品，当然这种产品未必是最终交付给用户，但也算是产品。所以它们应该有机的正确认识到它们都是软件的产品，而不是只是源代码是产品，这样容易造成代码和文档之间，代码之间对立的关系，老是这样会走到偏激的地方，我觉得软件产品是一个有机的整体，它只是不同的表现方式。

最后我谈一下质量保证的问题。因为我也看到，有一个观点或者叫是事实还是原则，就是说满足市场才是第一位，代码的质量好象不是特别重要，这不是原话，这本身就是矛盾，不应该将市场需求和产品质量对立起来。什么叫市场需求？它其中核心部分就应该是产品质量，因为任何市场需求最终还是一种个体的一种客户的需求，你要占领市场，你要满足用户，同时你应该想保住它，希望他买下一个版本或者买你新产品，你应该有质量高的产品吸引和保住客户，因为人们不愿意使用他已经知道的有缺陷的产品，所以不应该把市场和用户的需求和产品质量剥离开来谈。也可能我没理解它的想法。

产品的质量，我想应该说包含一些要素，比如说它包含质量的控制，质量控制从具体实现方法就是测试，像微软比较成功的就是无论是人员的一体，还是部门角色划分的区分都是测试比较严，它可以通过大量的方法可以把产品的缺陷及时地尽早地发现，并且加以解决。这是非常行之有效的。现在国内企业有认识到这一点，从没有测试部门到增加测试部门，当然比例不是很高，这测试部门不一定在行政上独立，但是角色一定要独立，绝对不要开发人员自己测试自己东西，这样效果不会很好。

另外从CMM角色来讲，质量的确保，它希望同时定义良好的方法，能够在整个开发过程中来实施这样一套过程和方法，从而保证你本身开发过程中产生的缺陷就比较少，这和质量控制还有一点缺欠，质量确保是希望能够利用好的方法来保证出来的产品本身就比较少的缺陷，因为零缺陷不可能。这样反过来讲，你再通过测试的努力，应该说会少一点。因为测试很好，但是大家要知道，如果你在系统测试或者比较靠后阶段发现的问题，如果它是在设计或者是需求或者编码引入的话，重复的劳动是非常大的。一个开发的组织如果这种重复劳动大的话，它的成本就上去了，我们的市场竞争力就在这儿，因为你最终产品定价可能市场已经决定了，你怎么把你的利润提高呢？那它决定你的效率高，你的充分劳动降低。这是我对质量的认识的两个方面。

提问：

有两个问题，第一个你曾经说到从需求开始一直到产品的推出，有一个可塑性的管理，这中间的一致性怎么保证。第二个问题，如果后期犯了错误，如果在开发初期就已经埋下祸根，如果避免开发初期的设计上的错误？

胡大庆：

我先回答第二个问题，质量确保就是这样子，它通过正确的方法，比如CMM规定等级的评审，他对你做出的文档，对代码阶段性成果都要通过一种方式来进行评估，通过这样工作就可以把本阶段，或者把本阶段以后错误发现出来，而不要带到下一阶段，这样就保证错误尽可能提早发现出来，错误越早发现出来，给你带来的充分的劳动工作量就越小。应该说质量确保这样机制和方法的引入可以适当的保证你不应该把更多的缺陷带到更多的测试阶段。

第一个问题关于可塑性，当然企业具体的做法不一定一致。比如说可以用一些工具，包括微软也可以提供一些工具，其他专门软件工程公司提供的工具，你版本的管理，从你项目计划开始，到分析、设计，到原代码的管理就是保持一致，就是目前你所做的设计你依据的是哪一个版本的需求，你下一代的原代码是不是依据这个需求，这有一个良好的跟踪观察和反馈的强有力的保证方法，还有靠制度和人员基本概念要培训，让大家都知道，你是在用方法做事情。另外还有一点比较重要，就是说这个概念国内大家也逐渐认识到这个问题，任何原代码即使你认为原代码很重要，大家应该教育工程师，你写完原代码，经过一个阶段，交到一个地方之后，这个东西就不是你的了，不是说只要你写的，就是你的，我随时可以拿出来，我想改就改，这就错了。如果是这样，你慢慢就会有一些错误发生，这样就不会知道谁是最好的，谁的设计和现代的是一致的，这样你把这种制度或者过程建立起来，而且教育员工一旦提交之后，这就是项目组的共同产品，然后再进行怎么修改，然后再怎么做。

提问：

我了解您是从摩托罗拉来的，像摩托罗拉关注什么软件，还有手机上有什么样的软件？

胡大庆：

第二个问题我没法回答，因为我们这个部门不是做手机的。我们部门是做系统的，你打手机实际上有很多机站在我们背后的高楼大厦里，如果没有这些机站，你的手机也没有信号，我们是做这些系统的。我们现在做的百分之百软件都是将来要嵌入到我们硬件设备里，无论是机站，还是交换机，还是其他设备中，所以没有任何软件推向商业化市场的。这跟大家区别比较大，基本上就是这样。

提问：

您刚才讲到质量保证，你是依靠一整套质量体系保证产品质量，我听一些朋友说，他们在小型的团队中他们应用的不是完整的质量体系，他们可能用（英文）的方法，满足客户的需求，以后出现问题时，我肯定有办法把它重构好，你说的完整质量体系对小公司是不是也实用，因为他更需要的快速的反应。

胡大庆：

我觉得有一个特点，虽然一个人做软件也要引入软件工程，但是人多和人少做软件还是不一样，我觉得一个五个人十个人做，十年五年都想保持这个，你可以这样。如果你想扩大你的软件企业的规模，我想你的想法你们就不一定能满足你企业继续发展的需要。如果你的市场是某个校园里的大学，他每个礼拜都让你出新东西，你没有必须保证质量，如果你满足这个小的需求就可以了。但是我们从我经验来讲，我们的软件比如中国移动它有上百万的用户，我不能奢望它能原谅我。所以确实不同规模，不同市场的需求下，质量的要求或者做法，或者这方面投入也是不太一样的。就是说在这个阶段你不可能百分之百解决好的方法，因为你没有这么大投入，因为你可用你的思想，引入到你的公司里面对你下一步发展是很好的。否则你发展到一百人的那一天你再想让大家改过来这个习惯就很难了。

提问：

摩托罗拉是用6个?了，你能解释一下？

回答：

摩托罗拉更主要的产品是大的机站，所以 6 个? 是追求整个质量的特征，我讲了 CMM 也不是十全十美，它对后面客户服务业没有涵盖住，在我们所在的部门也要用很多东西来弥补，因为我们有一个不太大的硬件开发队伍，还有机制上测试的东西，这些东西不能用 CMM 来涵盖，有人讲 CMM 全面质量控制管理在软件上的实现，对我们来讲 6 个? 对这个并不是矛盾，它应该相辅相成的有一些帮助。

提问：

大家都知道软件工程主要消除个人因素的影响，但是我觉得每个人的一些特点也是需要的，在微软和摩托罗拉当中对具体个人素质有什么要求。我知道 CMM 有 TSP 和 PSP（英文）。

胡大庆：

我们百分之百的员工在大学阶段都是在国内受的教育，我们的比例当中也有一些博士也有一些硕士。我觉得更多的是摩托罗拉做的比较好的一点，就是公司里的再教育，或者叫做 ON-Job Training，这确实至少在国内外企里做的相当不错。它把一些过于理论的东西根据企业的要求进一步广泛的培训。所以我刚才讲全参与全培训，你希望他做这样的事情的人你没有跟他讲清楚，你不可能奢望他有一个好的结果，因为本身人的惰性就很大。某种意义上就有点像公司的规章制度一样，我要求你这样做，不是可商量的。比如像项目管理，经理要求你开什么样的会，参加什么样的活动，出什么样报告这都是有严格要求的。但是这个是不是抹煞了个人的创造性，其实不是。反过来节省了你不必要的浪费，时间的精力，可以把更多的精力放在让你发明创造这方面。

提问：

软件质量有一个前提，如果对软件规模没有估算，我觉得可能会导致计划进度安排不太准确，可能达不到目的？

胡大庆：

在需求分析阶段也会写一些方法来付出，但是没有绝对一个方法保证你百分之百正确，比如你整体交付原代码的估算是多少，这个我想更多的是经验是非常重要的，或者更多的如果你原来有类似的项目做过，这都是一些依据，另外你把你的需求尽可能细化，细化到以周为单位的工作量，就是尽可能细化，这样比较准确估算出你最终用多少人多长时间，但这也是一个计划。我觉得 CMM 一个比较大的特点，我们的计划按照一定的要求是不断地在更改的，因为一年前定的计划明知道它不实用了，还为什么要留着它呢？但计划的修改还要得到一些人的认可。只有有针对性的，确实能指导你的计划或者需求才有继续往下做的意义，否则慢慢就扔掉了，没人看了。但是如果没有一个东西指导你，你凭脑子来想编程，你越往后你就会发现你绝对要失控的，因为我们在谈向规模化的企业。

杨永生：

我同意胡先生的观点。规模小的时候你可以控制，这是一，第二组织阶段来说做一件事情也好控制一些。反过来开发人员对日程的安排来做出最终的解释，他根据他的反馈很大程度上还根据经验。另外你永远要加一些融余，就是做一个实际的估算，因为可能市场的推广，全依赖于你按时把这个完成。还有作为 PM 时你要做一些取舍，到关键时候可能要砍掉一些东西，或者增加一些东西，主要是取决于你发布时期绝对不能改变的。

提问：

你说主要由开发人员来决定议事日程，这会不会导致产品开发时间的延长？

杨永生：

这不会，你需要做这些事情，开发人员会告诉你做一些什么，微软一些操作系统你可以看到，做这些事情哪些需要做哪些可以不做，作为取舍是很重要的，不要试图一次把所有问题都解决掉。像IE，IE要放在Window里面，这不仅仅是你的事情，可能Window做“BTA”(?),可能两个月前就不让你做了。

胡大庆：

还有一个大的框架，就是让开发人员来作主或者参与并不是让他做决定，同样的产品他知道大概花多长时间，如果全部由上面人来决定，你没有反映到底下，或者是上面完全压下来的，开发人员即使知道做不完他也不说，如果这样对公司也是不好的。所以在这个意义上，也是在这个基础上的。

提问：

您讲一下代码的复审问题？

杨永生：

代码的复审没有正常的过程，就是往往经验丰富的人做一些，到最后稳定化阶段，这时候你可以会做一些“瑞维沃”，测试人员、开发人员、项目经理都会看得到，而且都会详细的看完，或者给一些反馈。项目经理之间，虽然是一个PM，他有一个特性，但是有一些地方他会请别人看一下，对于全球化或者特定的东西，他会请一些挺懂的人看一下。

提问：

微软的开发开发方法，自《微软的秘密》之后有什么新的发展？

杨永生：

从产品开发过程来看还是那样，无非是角色的变动，基本原则并没有大的变化。

提问：

从您讲的微软的岗位设置来看，您认为中国在这些人员中最缺乏是人员是哪些？您认为怎样去解决？

杨永生：

我们最缺乏的人员我想在座的每个人自己回答这个问题更好一些。

提问：

刚才胡先生讲软件开发过程，所有文档都作为产品来看待，我不知道摩托罗拉对除了代码以后文档把它作为产品有没有一个规格？

胡大庆：

有，当然这个经验是不是可取，是不是大家都使用这是另外一回事。但我们做法基本上你要形成文档或者阶段性东西都是有定义的，都是有样本。比如公司要求你按照什么格式来写，通过什么样方式在一定范围内做一个评估之后，才能最终确定下来它是要继续执行来用的。我说得过程中的产品，一些东西，并不是说你在 A 4 纸上写了三行就算文档保存了，还是要稍微正规一点的东西。没有必要把文档都非常机械，非常像写论文那样来写，也没有那样的精力，但是你基本的要素，基本的东西还是要交代清楚，否则你就没有参考或者进一步利用的价值，因为每个东西既然是产品了，它对你下一阶段或者某些工作是有帮助的。

提问：

我想问一下，现在软件既然是工程，工程肯定有一些意外的情况，不是顺利的情况，最简单的像工期，或者预算严重超支，我想问一下几位，碰到这种情况，这种危机是怎么处理的？

胡大庆：

确实有这种情况，由于各种原因，比如我们有比较大的项目，它其中有一个比如叫功能模块或者一部分当时由于各种原因采取外包了，外包的子合同的承担确实因为各种原因没有按期交付给我们，那我们就没法把它加入到我们最终要提交的产品里，这最终导致了一个拖延，就是损害也是比较大的。

还有一个例子，我们有一个很小的产品，是一种在 O n l i n e 找一些记录，记录一些 O n l i n e 过程，同时判断一些错误。我们 1 5 人做了一年，做的还不错，但是最终用户也没有要。你说这个失败不失败？没有失败。你最终满足用户需求就可以了，用户为什么没有要？他可能认为这个没必要了，他认为有最新的方法，或者你下一代的产品已经替代他的需求了，但是在当时的市场还需要你来。从软件工程师来讲是挺可惜的，但从大的范围来讲还是有必要做。你不能说咱们做完了，他可能不要，那就不做，那肯定不行。

主持人：

我们最后一位主讲嘉宾是北京世纪豪杰计算机技术有限公司梁肇新总裁，大家都知道超级解霸软件是很有名的软件。我们知道软件最终要生成代码，代码质量往往决定了软件的质量，作为成功的程序员梁总可能有许多宝贵的经验。他今天发言的主题是《代码规范化》。有请梁肇新先生。

梁肇新：

我主要讲一些比较实际的东西，因为我们公司不是很大。我们 9 8 年成立到现在碰到了很多问题，在这些问题解决过程中，我总结出来代码规范化非常重要，因为从教科书上就没有一种规范化的提法。我发现很多开发人员最后写出来的代码都不相同，这样的话写出来代码就成了他自己能看，别人不能看，这样很不好。刚才讲到 C M M 里面，我想每个公司可能都有自己的规范，我就讲我们自己的代码规范。

代码可能有价值的，但是如果一个代码没有思想，就纯粹是乱码，是没有任何价值的。所以代

码分两种，要使得代码有价值，它可能要规范化，要有比较强劲的设计思想，代码本身要有思想，这样代码才有意义。以前包括有一些人到公司推销代码，我们一般不会理睬这种事情，因为他代码可能跟我们实际有很大的差距。所以怎么使得开发的思想统一，使得一个团队里面的行为能够一致。

我们实际的做法第一件事就是到我们公司的人员首先做代码规范化，我们做法是开发语言的统一，我们做一个相同软件我们不可能说一个用 DELPHI，一个用 V C，每个人写的都不相同，所以我们统一成用 V C 来开发，然后我们再要求格式，格式分为层次，就代码的格式层次一定要清晰。我们区分代码时用“TAB”，教科书里往往看到大括号不对应，很多“FOR”以后往往一行都是大括号，最后有一个解决办法就是在每一个括号后面写上是哪个括号的。

在处理代码，我们自己写代码的时候，我不知道现在开发人员怎么写，但是我们自己总结出来以后，我发现要写一个文件时要先把它主干提出来，然后再给它填枝叶，那怎么写呢？我们进行了成对编码，先写上大括号，再写下大括号，然后再加中间的，因为我们实际在以前做过的过程中发现很多代码最后分配了很多内存，没有地方去释放它，最后机器自然被消耗消耗，程序就死掉了，所以成对编码我认为就比较重要。

小一点程序在环境里面就自己生成，把它拼凑成一块就把它拿走，我认为这种方式不是很好，应该把它建立成 BUILD，把所有文件放在一起写成 TXT 文件，然后就输入 BUILD 命令，它就能把你所有你要的东西的相通的，所以（英文）比较重要。

我们实际发现使用过程中用 C V S 用代码管理比较不错，C V S 能解决两个人同时写一个文件时怎么办的，绝对不是说我编这个文件别人不能编，在同一个文件里头，也可以解决，两个人可以在同一个文件里写。这个 C V S 可以直接把这两个部分合并成一个文件，解决同时编码冲突的问题。

还有一些开发理念的习惯，我们觉得比如像代码里面教科书里面比较多的，还有实验性代码比较多的与 assert，我们实际使用中不要用这个东西，因为这个最后推出产品的时候，有两种方法，一种可能根本把这一行去掉，另外一种运行时出错，弹出一个对话框，如果你怀疑这个地方有问题，就应该有一个相应的处理方法在这个地方，就不要使用 assert 来取代它。

我们早期版本用过一些 MFC 的程序，后来我们发现错误很多，现在就不用了，MFC 可以产生很多废码，看起来效率很快，但是问题很多。我们宁愿慢一点，但是要很多东西都可靠，这些代码可运行性很高。在代码实际中分层次编码是非常重要的，在每个行数前面一定要有一个统一格式的注释，你代码里面没一个注释也是很重要的，但是每个函数前面有一个比较完整的注释，别人看这个代码时就像这个东西是我写的。这样使得这个团队有一个比较好的协作的基础。

主持人：

谢谢梁总的精采发言。

提问：

我想请问一下梁总对代码的重构有什么看法没有？它是指一开始的时候就比较快，还是比较规范的很快写出一些代码来，不考虑以后的需求变化，等到以后需求发生变化时再去给它想办法重整这个代码，在它不影响它的功能的情况下？

梁肇新：

我们使用动态库设计这些东西，在设计整个软件的时候，我们也有一些开发人员不了解这个开发性，代码写得比较封闭，一个程序用一个大类来实现，最后给这个东西添加功能时非常困难，所以我们就要求开发实际过程中尽量把所有东西能提出来放在动态库，以免将来改变时的麻烦，再不需要对动态库进行改变，你只要在需要改变的地方进行测试就可以了。

提问：

您以前是个很成功的程序员了，成立自己公司以后从开发第一线退出来，从第一线到开发人员这需要在思想上有什么样的转型。您对软件工程方面是什么见解？感觉到自己公司在软件工程专业上需要什么帮助？

梁肇新：

我们公司现在规模还比较小，我们最困惑的是2000年的时候，当时我们团队刚刚建立起来，为什么我们会建立代码规范化，也是因为实际中代码混乱，所以我们在实践中建立起来。年底的时候我们听说CMM，我们去请一些老师过来实际培训了一下，使我们视野开拓了一些，但是实际上还是按照自己实际需求，朝这个方向做是很有必要的，但是做的过程中我们是很谨慎的，不会把原来东西打乱了，我们主要目的是使我们开发更加可管一些，更加规范化一些。我自己一直定位为程序员，就是开发人员，而且我本身就喜欢编程，我觉得编程就是一种娱乐，而不是一种工作，我一直在做实际编码工作，所谓代码规范也是我实际使用中发现它有很多好处，然后再总结出来的。

提问：

您怎么处理编程的身份和总经理身份的？

梁肇新：

像我们公司技术方面我主要负责这方面，还有产品，如果要决定什么样的东西，比如2000年网络很火的时候，市场人员要求我们是不是也搞网站搞风险投资，我觉得这跟我们定位于软件企业根本不同，这两个相当于跨行的，所以这东西不是我们要做的，所以最终我们还没有做这方面的工作。我主要是做技术，在技术的前瞻性，看哪项技术能运用到哪些产品里面去，目前最新技术是什么，这些技术我们拿过来以后，我们能做什么样的产品，我主要考虑这些东西。而实际中我会做一些比较先进的底层的技术，做好后提供给实际项目中使用。

提问：

您是作为程序高手或者程序英雄，我非常好奇，如果您总是编程，您不可能管理好一个团队，您可能抽出时间进行管理，您怎样处理编程和管理这两件事情？

梁肇新：

管理不是我的长项，我们工作的日常管理和其他管理有其他人做，我就相当于技术骨干的角色。

提问：

我问一个问题。我接触了很多出版界的朋友，我发现国外软件工程软件管理在知识资源上非常丰富，但是国内比较少。那我想问一下，大家也是在实践中去带团队，也要鼓励自己团队中成员去成长。从知识角度来说，就看他们有什么样的素质，我希望嘉宾从亲身体会来说，如果您希望团队成长的话，从知识角度怎么样来配合？

周之英：

我想这个问题从我们国家来说，应该从教育和工业界共同努力来解决这个问题，某种意义上，从国际上包括美国它也是重新认识这个问题。从总体来说，软件工程是超越一个具体项目，但是很多人现在并没有把软件工程中间那些思想本质的东西掌握到，而是把它当成很简单的理解，一个规范，我能够照猫画虎就能做了。

我在学校里教课也碰到这样的矛盾，从我的领导一直到学生，都有这样的矛盾。就说他们老是企图，说周老师你教的东西太抽象了，我也学不了那么多，你只要告诉我一个样本，说我为什么来上你的课，我的老师要这个项目，他现在要写可行性报告，你有没有东西要抄一份，这是很典型的。这个问题可能特殊点，但是对大多数人也一些影响，程度不同而已，就没有把它作为很重要发展软件工业的基础来看，它里面有不同的层次。

从国外这方面的开展研究很多。我们国家一开始就有一个误区，就是从80年代初开始搞软件工程的时候，认为软件工程一个是规范，第二个中国发展出自己的工具来就行了，就把这个看成非常死的东西。一个规范，什么东西不需要，我就照着做就能解决了，第二个误区我开发出一个东西，然后这个东西就能解决一切的问题，实际上我们发展了二十年来，是在不断地探索，这样一个人怎么能够来做软件一些规律性的东西，技术的东西。然后在做工作里面有各种不同的层次，包括编码，然后到需求分析，到开发，然后发展新技术，有很多层次。所以这个东西我就觉得今后从教育的方面，比如教育就像编程问题，我们看没有一个教科书里面写工业编程是怎么编的，学的编程是学的语言，就把语言弄懂了，这是面向对象的语言，这是结构化的语言，是学语言而不是学工业化的东西。

其他层次的都是需要很深层次，我有时候也很困惑，不是计算机系的老师跟我说，我对软件一点不懂，我现在要搞一个项目，你告诉我我怎么去管。我想如果建造一个房子，没有一个人对建筑师的教授说，你告诉我，我明天要设计国家大剧院了，我对建筑什么也没学过，我这是人搞化学的，我明天设计国家大剧院，你告诉我什么办法，我明天去投标去。但是你看软件到处是这样的状况。当然软件也有它方便的，软件很容易编，但是你要做到不同层次的话，就有它很多深层次的问题。

所以我觉得一个从工业界怎么不断总结，一个从教育界怎么能够改变过去关心计算机科学技术而是关心软件工程的问题，以后可能对我们国家的长远发展会有一个比较深刻的改变。

主持人：

今天到现场的还有一位，大家知道国内软件工具有一个叫做“PLAYCASE”，这是一名叫高展的先生基本上是他一个人开发出来的，因为国内软件工程大家是谈的多，做的少，软件工具方面像青鸟开发了一整套，但是一般公司一般人都见不着，不知道用在什么地方，但是像高展先生，他一个人踏踏实实地把“PLAYCASE”写出来了。下面请高展先生说怎么把“PLAYCASE”写出来的。

高展：

书生公司王东临先生提了一堆问题，我从两个角度归纳了一下，更容易回答书生公司提出大概几十条的问题。这两个问题第一个问题是软件工程实践者的观念问题，主要是中国的软件工程实践者，再一个软件工程方面自己的原理上是不是也有问题。

首先我们中国软件工程实践者自己的观念问题，我前一段时间曾经发表过一篇文章，说中国软件公司是没管理没技术，大家可能不赞成这个观点，实际上现实情况是这样。软件工程就两大因素，一个是软件开发技术本身，像怎么做需求，怎么做编码；再一个管理问题，再一个看软件开发管理上。因为软件开发主体基本上是软件公司，公司非常明显的特色就是分工特别清晰，分工问题是在230年就提到这个问题了，我们的软件公司在去年的时候才提出，这个结论是百分之百的正确，所以我们软件公司观念落后了230年了。今年就不同了，有专门的部门了。再一个软件管理水平的确管理很差，就是量化管理。在传统行业上，我们整个管理水平也是落后大概一百年左右的水平。因为在二十世纪初的时候，有人把一个建筑工人在弯身、拿砖头、抹灰这三个动作进行分解，然后再进行测量，最后实行了精确管理。我们国内一般企业都做不了这个管理，不用说软化公司了。所以软件公司整体开发水平并不是特别乐观的，这主要是观念问题，这方面大专院所也做的也不是特别理想。所以这方面可能从观念上这方面问题更研究。就是大家首先认为需求不可能做透，好不容易认清楚了做了“coding”。

所以整个软件开发从生命周期来讲，我们每个阶段都在犯错误，所以我个人理解，我们现在软件开发，做管理软件来说在于需求，需求关键在于哪儿？我个人认为对于用户本身业务的建设，这问题最大。所以这里面里头涉及到软件工程原理的问题，从我个人从软件来讲，我应该是后外行，我不是学软件出身的。

我大概在89年90年接触软件工程。我们当时学，老半天很不容易学懂之后，发现很难为用，我们从书上，从国内外书上看作需求分析的方法，太难在工程当中进行操作使用，所以最后我总结出来，可能是软成工程这门书上对如何做需求分析可能给教错了，做设计做其他的可能都没有问题。现在最大的在源头可能有问题，因为软件开发源头被污染了，下流再怎么治理可能一点意义都没有。所以我个人认为软件开发可能是一个废品的概念，就照葫芦画瓢，这瓢就是软件，这葫芦就是应用领域，企业的管理模式，从根本上解决需求分析做的步骤，可能首先把依靠用户对领域的问题解决掉。这方面我们做的还是有一些成果的。我们提出来一个观点，全程一体化……就是对用户领域的分析，做需求分析，然后做概念设计再做总结设计，这样就会很顺畅。这样某种意义上消除了软件总体设计时间，包括详细设计时间也会消除到60到70%的时间。

这都是我个人的观点，一个感觉是本身中国软件工程的这些实践者观念有极大的问题，现在虽然这些问题在逐渐消灭掉，不过消灭也有很大问题，我大概在97年98年的时候，是我在跟软件公司在讲软件工程的重要性，到99年这些软件公司的总经理在给我讲软件工程给我们带来什么好处。我现在感到感觉最大的难处是如何说服具体开发人员认识软件工程的重要性，我现在做了很多培训工作，绝大部分都是总经理技术总监，包括部门经理来推动这件事情，唯一遇到一个案例就是员工做这件事情，总经理不愿意干，可能怕花钱，员工学会了会干别的事情。这是观念问题。至于你采取什么方法来做，这都是无所谓，主要是观念问题，大家认为软件开发能不能按流水线方式来做，这是最重要的。

提问：

全程建模我以前也用过，这有一个问题，这是一个普遍的问题。这是关于UML的，在你的软件里面我也用过UML，在使用的UML和面向对象的时候你是怎么实现的？UML在整个软件工程里面它起到什么作用，主要在设计上面的作用？

高展：

用我们一体化如何来体现UML方式，我们做了一些工作，就是我们把面向对象和结构化这两个做了一个综合，所以实际上我们兼容了大部分UML，包括它的顺序图，包括对象图，这是比较主要的，在我们这里都会得到体现。它的图形语言我感觉我们大概兼容了80%，就是一一对应的方式。

包括对象应用，跟结构化没有区别。这个结构化对象分解本身就是结构问题，就是对象组成关系本身就是结构问题，所以这两个完全没有区别。

提问：

因为我在用你的Playcase建模的时候，我发现还是以模块化为主，就是面向对象很难加进去，实际上我们软件公司要求软件编码的复用要求程度很高的，所以我觉得你在这方面有一些缺陷或者不足。

高展：

这应该看办法和工具，从办法学角度来讲我们是没问题的，如果你用纸和笔来肯定不会出现这个问题，可能我们工具在这方面有一些问题。

还有一点我稍微解释一下，关于模块和对象的问题，我个人的理解，软件工程所有的追求目标就是模块化，所以实现模块化的手段构建对象都大同小异，包括用结构来描述，这是我个人的观点。

周之英：

关于结构化，我想软件工程的最根本的问题，就是要有一个最主要的就是控制复杂性，就是说你的规模特别复杂，整个问题很复杂，所以要分解，不管怎么样都是用一种分解的方法，就是把大的问题变成小的问题来分别考虑，考虑它们之间的关系，这是一种不同的出发点。从这个意义上对象之间的关系也是一种结构，只不过这种结构看你怎么理解，过去的结构化方法主要是指模块调用图这个结构，现在更深层次我觉得这些东西都解决不了一些复杂的问题，具体说起来就是现在提出的非功能性需求，比方说性能问题，安全问题，这些问题为什么不能够解决，按照目前来看没办法分解到某个模块当中，所以这问题就变成非常困难的问题，如果单纯功能性的分解，某种意义上不复杂，这本身你能分解的。为什么大家觉得就开发到现在的用的工具来开发一些管理软件或者什么，你拿上来了以后提更高的要求，不复杂，因为你变成的语言可以很高级了，很快就可以学会了，然后把几个需求划分了以后大家一起来。但是你分散了一致性，安全，再加上黑客攻不攻，性能要快，各式各样的问题，这些问题加进去问题就复杂了。所以用各式各样的方式去解决。

还有UML，我想它是很好地刻画了面向对象的東西，通过（英文）组织也是很有组织的一大群，跟CMM也是差不多类似的这种规模的状态。提出各种需求，比方说package包装的要求，各种部件之间接口的关系，它会产生什么样的需求，所以它是从本质上非常丰富，可以刻画这些东西。建模无非是很好的描述手段，但是描述手段并不等于你能够解决问题，因为你要建什么模型还是要靠人去解的。所以并不是说UML来了以后我就行了，它只是代表现在很多领域提出的问题，它怎么来表达，它最基本的实际上就是表达相关关系，集成关系，这是最本质的东西，也就是UML最早的那点东西是最基本的。假如你要更丰富一些，比如大系统控制很细，你就可以把很细节的东西去描述起来，一描述了也来了一个问题，你描述的越清楚信息

量就越多，主要问题就不突出，可能带来了其他方面的问题。所以不要觉得它很丰富就一定很好，你也得要根据你的问题重点去选择。

第二个就是说UML并不是说你可以睡觉就可以解决问题了。根据问题的复杂性你要去建模这是根本的，这是一个根本的语言。

主持人：

谢谢周老师，我们今天的沙龙就到此结束，谢谢大家的光临。谢谢。

[本文转载自新浪网]