美国和日本软件开发实践的比较

软件开发作为企业信息系统的重要部分受到信息系统管理人员,分析员和研究人员的高度重视。生产高质量软件不仅对用户的信息技术管理有重要作用,而且是软件开发公司竞争优势的主要来源。日本的软件产业在历史上落后于美国。可是,近几年日本硬件的发展是沿着在生产效率和质量上大大改进软件的道路前进的。

日本软件产业同美国相比。美国的软件开发还能继续优于日本吗?日本取得软件质量和生产效率的主要因素是什么呢?什么是制约日本软件产业发展的限制因素呢?在软件开发上美国对日本有什么竞争优势呢?

对于这些问题的解答并不一致。有人断言,尽管日本人取得了进展,而美国将继续支配软件领域。也有人认为,美国占优势的时代已经过去,就象在计算机硬件和汽车制造方面一样,日本软件公司已成为美国对手的严重挑战。还有人得出结论说,日本与美国的软件性能之间没有重大区别。

这些互相矛盾的看法主要是由于这两个国家软件开发实践方面的不一致性。例如,软件生产效率按照非注解性源代码行数(SLOC)做比较必须在项目规模和程序设计小组结构的范围内进行。同样地,只对软件的少数方面做结论,如开发工具(如 CASE)和面向目标方法学,而不把软件市场结构和程序员的工作效率一起在总体上加以考虑就可能误入歧途。

组织机构

日本工业的成功在于一方面保持内部效率的平衡,另一方面同产业内的其他参加者合作。这种倾向在日本软件产业上也反映出来。为了获得高质量和生产效率。软件开发公司是在"软件工厂"概念的基础上建立的。这些公司的工作通过各财团支持的全国性项目得以加强和协调。而且为开发、维护和升级提供长期支持的硬件公司的作用是不可缺少的。表 1 是美国和日本软件开发组织机构的比较。

表 1　美国和日本软件开发的组织机构

要　素	美　　国	日　　本
软件公司	小房间:程序员认定的个人生产中心,主程序员小组	软件工厂:遵循生产型组织,多级结构
财　团	小机构;害怕反垄断法律几乎没有政府支持	大公司财团规定通用软件标准,主要由政府组织和资助
硬件公司	硬件与软件公司是独立的,用户公司在软件开发中可能得到硬件公司的支持	硬件与软件公司之间联系紧密,硬件公司必须帮助用户公司开发自己的软件

R.W.BEMER 给软件工厂概念下的定义是促进生产效率和质量的测定与控制以及使用标准化工具的程序设计环境。这对于使用传统技术生产软件的团体来说是重要的。SYSTEMS DEVELOPMENT 公司是 1975 年首家建立软件工厂的美国公司,可是到 1978 年就停止了。软件工厂结构被用在美国其它软件开发项目上,如 HUGHES 飞机软件工程环境和加州大学的系统工厂。另一方面,日本许多公司把重点放在软件工厂结构上。

在日本,软件开发主要在容纳上百名编码程序员和百台终端的大房间里进行。而在美国,软件生产是由要求免除时间和其它限制的程序员执行的。日本典型的软件工厂是允许软件生产组织以系统方式设计、编程、测试、安装和维护商业软件产品的环境。例如,东芝的府中软件工厂(TFSF)是东芝公司的五个软件工厂之一。在 TFSF,2000 多名程序员在一个大型工厂式环境中编码和调试程序。东芝公司宣称故障率已减少到每 1000 行代码 0.3 个。这样惊人的低故障率使东芝公司能够提供 10 年担保书,在 10 年内发现任何错误免费供货。

软件开发公司的机构。在日本,软件工厂项目由部门管理人员负责组织。工厂内有许多个部门。虽然每个项目属于不同的部门。可是整个软件工厂遵循同样的管理程序。软件工厂的实施为日本人在开发速度,低成本

和高质量软件上提供优势。美国 IBM 公司采用的主程序员小组的方法日本并不使用。日本的多数公司选择分级小组结构。

财团。日本有几个全国性项目,加强软件工程环境。这些工作得到日本许多公司财团的支持,成果一般共享。有代表性的项目包括:

软件维护工程实验室(SMEF)的主要目的是为有高可维护性的软件实现更好的开发环境。SMEF 是 5 年期的项目,由日本主要软件公司的联合机构··联合系统开发(JSD)公司负责管理。

软件工业化生成程序和维护辅助程序(SIGMA)是 SMEF 项目的后继。它是 180 多家公司共同建立的大量生产高质量软件的基础结构。这个项目的最终目标是通过制造而不是手工劳动生产软件,把软件业从劳动密集型转向知识密集型产业。SIGMA 也是 2 年期的项目,由日本政府国际贸易和工业省(MITI)组织实施。

1985 年开始的 FASET 也是 5 年期项目,由 JSD 公司管理。它的主要目的是研究形式规范技术。看这种技术能否实际使用。研究人员试图开发一种软件工程师所能使用几种形式规范的样机环境。

公司财团对日本软件开发努力的成功有重要作用。相比之下,美国在软件开发上的产业合作是很有限的。美国 IBM 和 MICROSOPT 公司的双边协议是难得的。害怕反垄断法是美国其它主要计算机公司合作的障碍。不过,最近作为对日本威胁的反应,建立了几家研究公司。例如,MCC 公司是对日本"第 2 代"计算机项目的反应。软件技术研究是 MCC 公司的几种活动之一。

硬件公司。日本软件公司对硬件公司的独立性远不如美国为了维护竞争地位,日本硬件公司对用户软件的开发、维护和升级有重要作用。日本软件公司常常作为硬件公司的子合同户工作。

外部环境

日本与美国软件项目的区别必须考虑到环境因素,包括不同的企业文化、政府对产业的支持作用和软件市场。表 2 列出了美国与日本软件开发项目外部环境的比较。

表 2 美国与日本软件开发的外部环境比较

要 素	美 国	日 本
企业文化	雇员与公司之间很少有长期关系,高级程序员调动	雇员与公司之间普遍建立终生关系,程序员把全部精力用于一家公司
政 府	一般不予干涉和几乎没有资助	通过 MITI 积极参与大量资助
市 场	最大的软件市场,组装软件领先	市场占有率在扩大,组装软件市场在发展,不过仍不象定做软件那样普及

企业文化。美国与日本企业文化之间的差别对软件开发实践的影响比其它产业更深刻。这些差异使美国系统同日本软件的结合比美国发动机同日本汽车的结合困难得多。语言障碍是日本软件公司在国际上竞争的一个限制因素。另一方面,日本职业安全和忠诚的情况使得在培训程序员上的投资很有效益。与日本相反,在美国雇用和培训程序员是一种冒险,因为程序员离开公司去参加竞争有很好的机会。而且美国和日本的质量意识在软件开发中实际上有不同的含意。美国软件的主要目标是遵守规定的系统标准,而在日本,用户满意有更大的取向。因此,美国程序员在编码上比日本人(较多的时间用在程序规范和设计上)花更多的时间。

政府。推动日本软件发展的主要动力是政府 MITI。除了 SIGMA 项目之外,MITI 还组织了第 5 代计算机和 TRON 计划。计算机大公司协调努力开发自己的操作系统。而美国政府与软件开发和研究只限制在通过国家科学基金会(NSF)提款的活动上。MCC 完全由 20 家大公司资助,包括 CONTROL DATA、DIGITAL 和 HEWLETT-PACKARD 等公司。

市场。美国和日本享有实际软件市场。作为世界最大软件市场的美国在组合软件上比日本有优势。而且由于英文作为国际语言给美国在市场上以竞争的优势。可是,由于在效率、压缩、准确和文件编制上的改进,日本似乎大大缩短了这种距离。

开发工具和方法学

软件开发实践中使用的主要工具和技术包括软件项目管理与控制、CASE 工具、面向目标编程和重用代码。

表3是美国和日本软件开发工具和方法学的比较。

表3　美国和日本软件开发工具和方法学比较

要　素	美　　国	日　　本
项目管理与控制	成本与质量之间往往折衷较高的障率,编码占软件开发时间很大比重	质量往往是唯一标准,低故障率,在软件规范和设计上占较多时间,编码占时间较少
CASE工具	适度使用自动流程图代码生成,程序和重用代码程序库	高度使用自动流程图代码生成,程序和重用代码程序库
面向目标编程	接受和使用迅速增长	与美国相同
重用代码	主要公司接受概念	日本强制执行;目标为重用代码占50%

项目管理与质量控制。在日本,质量意味着用户满意的程度。在美国,质量被定义为遵守规范。日本的质量控制涉及"计划、生产、检查和操作"的控制周期。随着时间的进展,他们把软件质量看作标准的动态过程。质量标准不断修改和完善。如果6个月之内标准未作修改,那么就意味着无人使用这种标准了。

日本人在防止重复故障问题上的谨慎态度,是他们软件故障率低的主要原因。美国的平均故障率为每1000行源代码3个故障,比日本0.3的故障率高10倍。可是,不均等性非常高,虽然有关数据表明美国的故障率较高,而就故障率来说,美国和日本软件质量之间没有重大差别。日本软件质量的进展似乎是由于强调质量标准管理和使用工业工程生产方法。

CASE工具。一些使分析和设计阶段自动化的CASE工具称为前端工具。这些工具在图形和文本上能为分析员和终端用户记录、显示和分析关键字概念。使编程阶段自动化的工具称为后端CASE工具。这些工具能生成高级语言或第4代语言的源代码或目标代码。第3类CASE工具支持整个系统开发周期的所有阶段,称之为全周期工具。

CASE工具在美国和日本程序员中都广泛使用,用于软件开发的设计支持、自动流程图、数据管理、代码生成程序、性能测试和问题跟踪等不同阶段。据悉,日本软件项目尤其在自动流程图、编码和重用代码程序库上广泛使用CASE工具。可是,到目前为止,美国和日本软件开发实践之间的差异性在CASE使用上仍然存在。

面向目标编程。使用结构化编程技术使过去几年美国和日本软件产业有可能提高程序员的生产效率。可是最近使用面向目标编程(OOP)方法的骤增在结构清晰性、可维护性和程序重用性方面产生了重要利益。尽管OOP概念是60年代后期出现的,70年代后期作为SMALLTALK语言的主要部分使用。日本人很快评定了OOP的意义,正如CASE工具的情况一样。使用OOP的水平在这两个国家之间近似,并未构成一种区别的因素。

重用代码。日本程序员认为,软件开发的经验能够重用。例如,当你规定与以前开发的系统十分相似的系统时,现有程序的规范说明可以重用。经常使用的编程结构包括标准化的和列入"可重用模式模块"程序库的参数部分。这些结构都能再调用和同参数变量以及其它模式模块结合起来生成新程序。

在商业应用中,尤其是普通批式处理中,从20-30个标准程序模式中对数据名称和关键位置作一点修改能获得这种程序的90%。设计员首先选择适当的程序模式。修改数据名称和关键位置,并附加必要的过程。

日本人认识到,没有重用元件不能获得软件生产效率的提高,以系统或数据库为主的知识是重用知识的聚合。日本人相信,软件产生效率提高的50%能通过重用以前开发的元件和知识获得。在东芝公司的TFSF,以前程序代码的65%被重用。由于重用代码结果。平均每个程序员的生产效率提高到每月2000行代码,是美国程序员生产效率的几倍。重用代码的使用也有助于提高软件质量,现在东芝公司提供10年的保证期。

为了更有效地重用软件元件,日本人正在设计文件编制和重用指南手册。例如,东芝公司的程序员能使用软件目录,为其提供3级检索:个人用、项目用和整个文件。给程序元件标上说明,以便使软件工程师能决定哪些元件适合新程序。为元件的使用开发计算机辅助程序有助于程序员找到合用的元件。

(柯新编译)