

PSP 与 TSP 介绍

PSP、TSP 与 CMMI：发展历史

CMM 与 PSP

在 20 世纪 80 年代后期到 90 年代早期，SEI 开发了能力成熟度模型（CMM），为软件开发总结了组织级的最佳实践。SEI 特别会员 Watts Humphrey 决定将 CMM 的基本原理应用于单个开发人员的软件开发实践中。个人软件过程（PSP）就是他努力的成果，为单个软件开发人员设计的 CMM 五级过程。

CMM 与 TSP

不久大家就发现，虽然使用 PSP 可以取得优异的结果，但是如果周围的环境不能鼓励并且要求遵守 PSP 实践，这些必要的规范性是几乎不可能得到维持的。所以汉弗莱为大多数组织中最小的运作单位——项目组，开发了团队软件过程（TSP），TSP 是为项目组设计的 CMM5 级过程。在一份 [SEI 技术报告](#) 中，记录了使用 TSP 在满足成本和进度估算的同时达到一流质量水平的最新结果

CMM 的演变

同时，CMM 的成功也引发了相似模型的开发以覆盖系统工程（SE-CMM）、集成产品开发（IPD-CMM）、软件采购（SA-CMM）以及人力资源（P-CMM）。为了缓解模型数量的过快增长，SEI 使用从未正式发布的 CMM 第二版、系统工程 CMM 和 IPD-CMM 开发了[能力成熟度模型集成（CMMI）](#)，

关于 PSP

何为 PSP？

个人软件过程（PSP）向工程师显示 如何

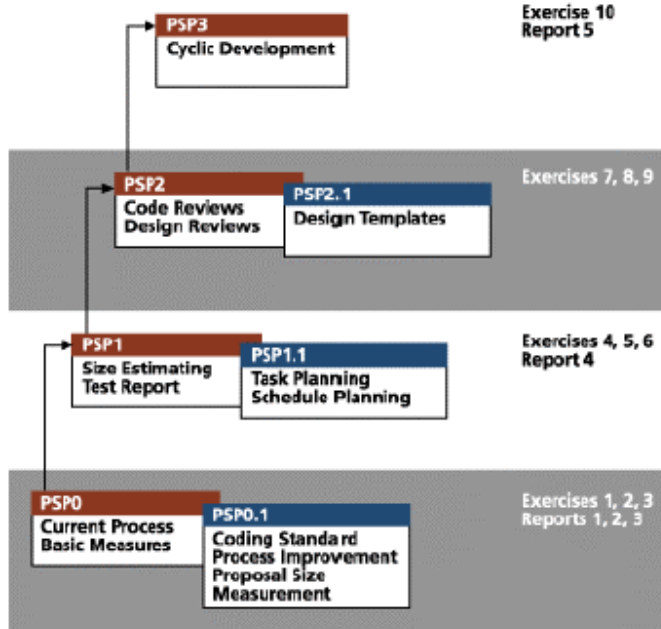
- 管理其项目质量
- 做出可以实现的承诺
- 改进估算与计划
- 减少产品缺陷

由于人员成本占据了软件开发的 70%，所以工程师的技能与工作习惯很大程度决定了软件开发过程的结果，[基于 CMM 中发现的实践](#)，PSP 可以被工程师作为指导，帮助建立开发软件的一套结构化和规范的方法。PSP 是组织计划引入 TSP 的前提条件。

PSP 可以被应用于软件开发过程的许多方面，包括

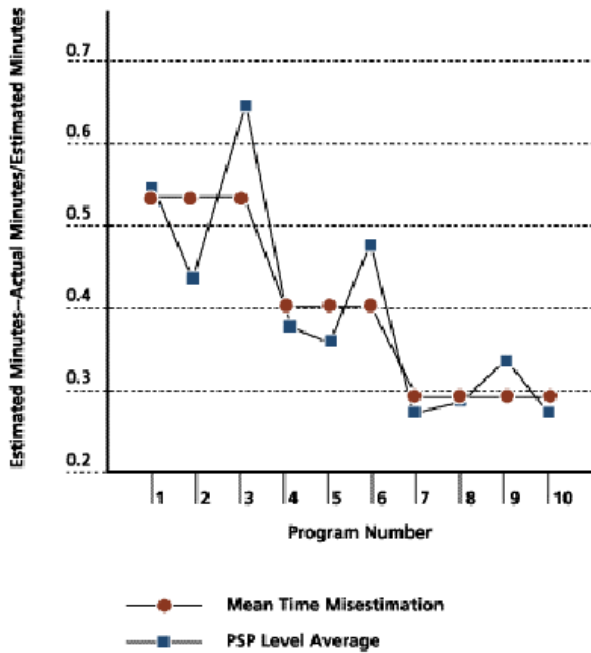
- 小型程序开发
- 需求定义
- 文档编写
- 系统测试
- 系统维护
- 大型软件系统的加强

图一：PSP过程的演化

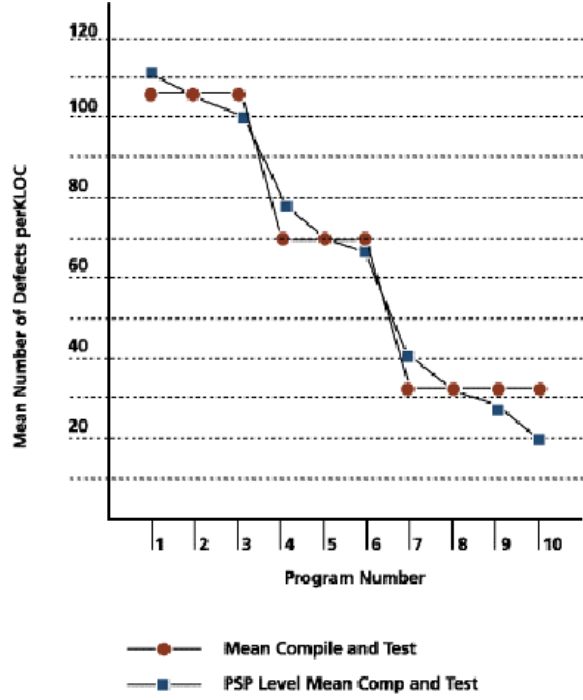


图二到图四显示了工程师经历的部分收益。图二显示了估算偏差从55%降到27%约两倍的改进。如图三所示，编译和测试缺陷的改进最为显著。从PSP0级到PSP3级，工程师的编译和测试缺陷从每千行代码110个缺陷降低到20个缺陷，超过5倍。图四显示了尽管计划和绩效有了显著改进，工程师的生产率基本保持常数。

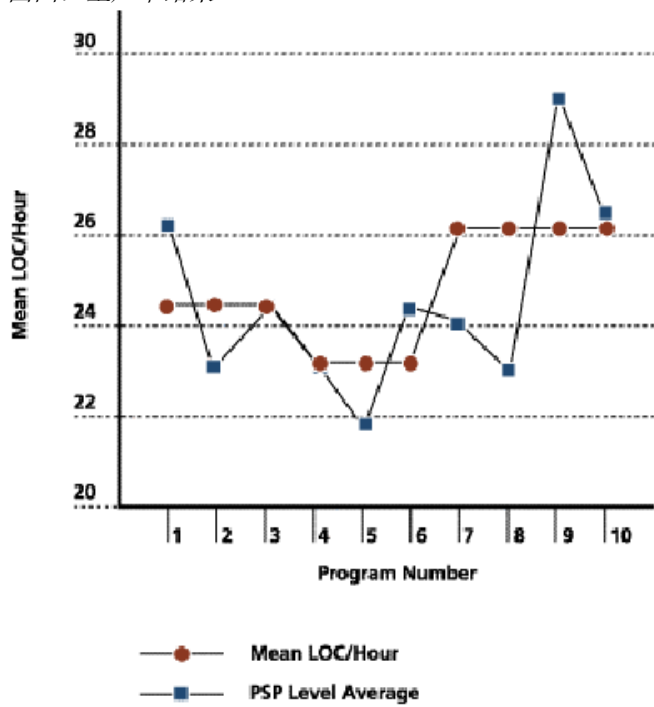
图二：工作量估算结果



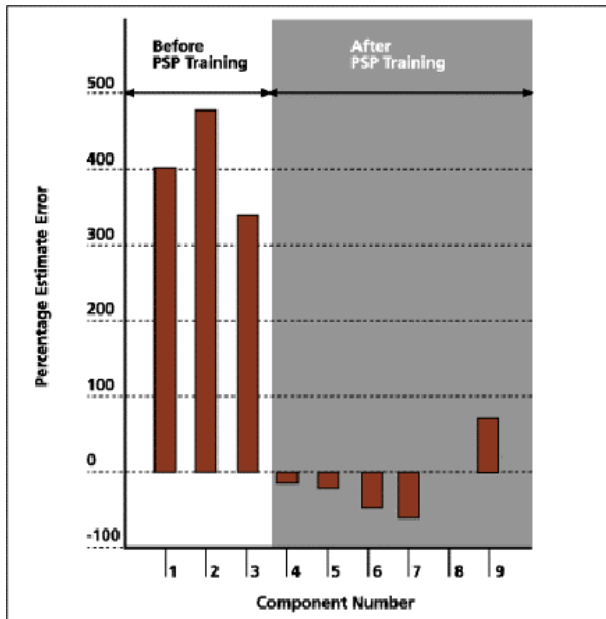
图三：质量结果



图四：生产率结果



图五：进度估算错误



PSP 的行业结果

使用PSP的组织数量正在增长，比如微软、Baan、波音、摩托罗拉以及Teradyne. 来自于早期使用者的数据清楚地展示出PSP培训的收益。图五显示的是来自于佩奥利亚的高级信息服务团队（AIS）的数据。该团队的成员在项目进行中接受了PSP培训。表格左边三个属性显示的是工程师估算前三个组件开发所需的星期数。例如组件1，原来的估算是4周，但实际花了20周。平均估算偏差为394%。PSP培训之后，同一批工程师完成了剩余6个组件。如右边所示，平均估算偏差为-10.6%。比如组件8，原始估算为14周，实际工作正是在14周内完成。

表格1显示了AIS工程师的一个组产品验收测试的数据。在PSP培训之前，他们验收测试的缺陷数很大，而且产品一直延误。在PSP培训之后，下一个产品基本按期交付而且只有一个验收测试缺陷。表格2显示的是9个PSP项目系统测试节约的时间。在表格顶端，显示了几个产品在PSP培训之前的系统测试时间。底端显示的是同一组AIS工程师在PSP培训之后完成产品的系统测试时间。注意A1和A2是同一产品的两个部分，所以对于它们的测试是在一个半月内一起完成的

表 1 Acceptance Test Improvement

Not Using PSP	KLOC	Months Late	Acceptance Test Defects
1	24.6	9	N/A
2	20.8	4	168
3	19.9	3	21
4	13.4	8+	53
5	4.5	8+	25
Using PSP	KLOC	Months Late	Acceptance Test Defects
1	22.9	1	1

System test time before PSP training

Project	Size	Test Time
A1	15,800 LOC	1.5 months
C	19 requirements	3 test cycles
D	30 requirements	2 months
H	30 requirements	2 months

System test time after PSP training

Project	Size	Test Time
A2	11,700 LOC	1.5 months
B	24 requirements	5 days
E	2,300 LOC	2 days
F	1,400 LOC	4 days
G	6,200 LOC	4 days
I	13,300 LOC	2 days

Table 2: System test time savings

导入 PSP

尽管对PSP的导入可以进行得很快，但也必须做得正确。首先，工程师需要得到合格的PSP讲师培训。SEI会自己培训或授权PSP讲师提供数量有限的现场PSP培训。也有越来越多的经过SEI培训的

PSP讲师提供商业化的PSP培训。潘树仁，循序咨询客户服务总监，是唯一长驻中国的合格的PSP讲师。

PSP导入的第二个重要步骤是分组培训。当组织要求志愿者参加PSP培训时，他们只得到PSP技能的冰山一角，一般对任何项目的绩效都没有什么影响。

第三，有效的PSP导入要求强有力的管理支持。这反过来要求管理层理解PSP，知道他们的工作人员一旦受训后如何支持他们，并定期监督他们的绩效。没有恰当的管理层的支持，很多工程师逐渐又回到他们的老习惯。问题是软件经理，象大多数的专业人士一样，当无人注意或关心时，发现很难持续进行规范的工作。软件工程师需要定期的指导和支持以保持高水平的个人绩效。

最后的问题是即使当一组工程师都是接受过PSP培训的并得到恰当的支持，他们还是需要找出如何将他们的个人过程组合到整个团队过程中去的方法。即使在高一些的CMMI级别，这也会是一个问题。这也就是为什么SEI开发团队软件过程TSP的原因

关于 TSP

什么是 TSP?

团队软件过程(TSP), 和个人软件过程结合在一起，帮助高绩效工程师来

- 确保高质量软件产品
- 生产安全的软件产品
- 改进组织的过程管理

工程组使用 TSP 将集成团队概念应用到软件密集系统的开发中。一个 4 天的启动过程帮助团队和他们的经理

- 建立目标
- 定义团队角色
- 评估风险
- 制订团队计划

在启动后，TSP 提供一个定义的过程框架用以管理、跟踪和报告团队过程

使用 TSP, 组织可以建立自我导向的团队，计划并跟踪他们的工作，建立目标，并拥有他们的过程和计划。可以是纯软件团队或集成产品团队，3 到 20 个工程师

TSP 将帮助你的组织建立一个成熟规范的工程实践，生产安全、可靠的软件。对软件买家和开发商，TSP 也作为一个新度量框架的基础得到使用

团队软件过程(TSP)扩展和完善CMMI和PSP方法，指导工程师开发和维护团队的工作。它展示如何建立一个自我导向的团队及如何扮演一个有效团队成员的角色。它还向管理层展示如何指导和支持这些团队和如何维护一个培养团队高绩效的环境。TSP有5个目标：

- 建立自我导向的团队，计划和跟踪他们的工作，建立目标，并拥有过程和计划。可以是纯软件团队或集成产品团队(IPT)，3到20个工程师
- 向管理者显示如何教导和鼓励团队和如何帮助他们维持颠峰状态的绩效
- 通过使CMM5级行为成为正常和可预期的行为从而加速软件过程改进
- 为高成熟度组织提供改进指南

- 支持行业团队技能的大学教学

TSP的根本好处是它向工程师展示在成本控制和大胆的进度计划下如何生产高质量的产品。它通过向工程师展示如何管理自己的工作，并使他们成为计划和过程的主人来实现这一点

TSP 的行业结果

TSP 帮助团队表现得职业化

也许TSP最强大的效果是可以帮助团队管理其工作环境，产品组遇到的最常见问题是不合理的进度压力。尽管这很正常，也具有破坏性。当团队被迫按照不合理的进度工作时，他们无法制订有用的计划。他们生成的每份计划都不符合管理层命令的进度要求因此不能接受。结果是他们必须在没有一个有序的计划指导下进行工作。在这样的条件下团队通常要比其他时候花费更多的时间以完成项目。

TSP团队的职责是尽可能迅速并且有效地制定计划并生产一个高质量产品。相对地，管理层的职责是及时启动项目并根据需要完成。当相似的项目花了18个月而管理层却要求9个月完成，这明显是不实际的。9个月之前项目本应启动的时候管理层在哪里？尽管的确有实际的商业需要，项目组的进度安排只是问题的一部分。在这些情况下，管理层和项目组共同工作并合理确定最有效的行动方案至关重要。这通常包括增加资源，周期性计划或者尽早关注高风险组件。

虽然TSP团队必须考虑加快工作速度的每一种合理方式，最新的分析显示，他们也必须为自己的计划辩护并拒绝无法实现的指令。如果管理层想要改变工作范围和资源或建议其他方法，项目组将很愿意制订一份新的计划。很多团队发现TSP给他们提供了有力的数据以显示他们的计划虽然激进但可以实现。

TSP 在微软的导入结果

微软，作为全球最大的软件供应商，最近有一个项目试运行了 SEI 个人软件过程(PSP)和团队软件过程(TSP)，使一个软件开发团队改变行为、改进过程、从而交付更好的软件。

TSP 塑造团队。在使用 TSP 之前，该团队有 10 个开发人员，他们对项目、工作、甚至彼此之间感到灰心丧气。该团队的成员各自为战，而不是作为一个团队。在使用了 4 天后，这组人成为了一个真正的团队。

TSP 降低缺陷，改进质量。在 TSP 培训中，微软开发人员的单体测试缺陷从超过 25 个缺陷/千行代码显著降低到 7 个缺陷/千行代码。微软的团队，如其他的软件开发团队一样，耗费 40—60% 的整体开发时间在测试上，因为他们用这些时间来发现并解决产品的缺陷

但是，由于微软的 TSP 试运行团队花时间在早期的缺陷移除活动上，如个人评审和团队检查，他们的测试只用了整个项目工作量的 11.5%。最终，该试运行项目组按时将产品交付给了测试，并且是高质量的。这使得项目节省了 35% 的成本

导入 TSP

总的来说，当一组工程师启动一个项目时，他们得不到如何开展工作的指导。如果他们幸运的话，他们的管理者或1到2个有经验的工程师将工作在一个运作良好的团队，并有一些如何开展工作的想法。在大多数情况下，团队不得不仓惶应付一系列问题。以下是一些所有软件团队都必须处理的问题：

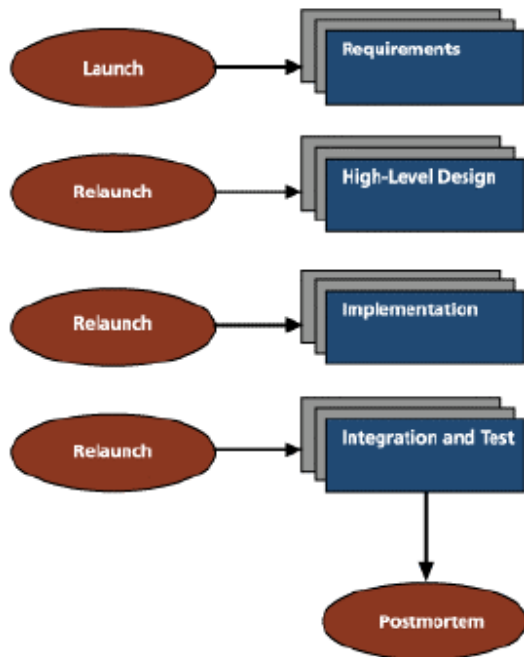
- 我们的目标是什么？
- 团队的角色有哪些，谁会扮演这些角色？
- 这些角色的职责是什么？
- 这些团队将如何作决策和解决问题？
- 团队需要哪些标准和规程，我们如何建立？
- 我们的质量目标有哪些？
- 我们如何跟踪质量绩效，如果出现问题我们应当怎么做？

- 我们应当用什么过程来开发产品？
- 我们的开发策略应当是什么？
- 我们应当如何设计？
- 我们应当如何集成和测试产品？
- 我们如何制订我们的开发计划？
- 我们如何使开发进度表最小化？
- 如果我们的计划没有满足管理层目标，我们该怎么做？
- 我们如何评价、跟踪和管理项目风险？
- 我们如何确定项目状态？
- 我们如何将状态报告给管理层和客户？

大多数的团队将大量时间和创造力浪费在这些问题上。这很不幸，因为这些问题没有一个是新的，并且每个问题都有已知和被证明了的的答案

TSP向团队项目提供明确的指南，指导如何实现他们的目标。如图6所示，TSP通过项目四个典型的阶段指导团队。这些项目可能在任何阶段启动或结束，他们也可能从开始到结束。在每个阶段之前，团队进行完整的启动或重启动，它们计划和组织工作。总的来说，一旦团队成员得到PSP的培训，一个4天的启动实践演练提供足够的指南供团队完成一个完整的项目阶段。团队需要一个两天的的重启动实践演练以启动第二阶段和以后的阶段。这些启动不是培训；他们是项目的一部分

图6：TSP结构



启动过程

为了启动TSP项目，启动过程脚本指导团队经历以下步骤：

- 和管理层评审项目目标
- 建立团队角色
- 对团队目标达成一致并文档化
- 生成一个总体的开发策略
- 定义团队开发过程

- 计划所需的支持设备
- 为整个项目制订一个开发计划
- 制订一个质量计划并设置质量目标
- 为每个工程师针对下个阶段制订详细的计划
- 将个人计划合并到团队计划
- 重新平衡团队工作负荷以实现整体进度表最小化
- 评估项目风险并为每个关键风险分配跟踪责任
- 进行启动后的检查

在最后的启动步骤，团队和它的管理层评审其计划和项目的关键风险。一旦项目启动，团队进行每周团队会议，并定期向管理层和客户报告状态

- 书面的团队目标
- 已定义的团队角色
- 一个过程开发计划
- 团队质量计划
- 项目支持计划
- 一个整体开发计划和进度表
- 对每个工程师的详细的下一阶段计划
- 一个项目的风险评价
- 项目的状态报告

如何延续TSP实施的效果：TSP管理者-教练

也许对于复杂而富于挑战的工作来说最严重的问题是如何保持规范性从而持续最佳表现。例如在体育运动或表演艺术里，我们很早就意识到需要高水平的训练师、指挥和导演。他们的工作是激励并指导表演者并坚持要求大家遵守很高的个人准则。尽管高水平的运动员很关键，但真正让队伍保持胜利的是教练。软件团队和运动队或艺术团体有很多区别，但他们都有一个共同的需要，即保持优异的表现。这要求教导和支持。

软件管理者没有扮演教练的传统，但这是他们在TSP中的角色。管理者的工作是提供资源、与更高层管理层的接口以及解决问题。但最重要的是，管理层必须激励队伍并保持对质量和优异性的不懈关注。这要求每天与项目组进行互动，并且绝对要求过程得到遵守、数据得到收集、结果得到分析。有了这些数据，管理层与项目组可以定期会面评价表现并确保其工作满足优异性的标准。

结论

CMMI、PSP和TSP为过程改进提供了一个三维的框架。如表三所示，CMMI有18个关键过程域，PSP和TSP指导工程师满足了绝大部分领域。这些方法不仅帮助工程师更加有效而且为加速组织过程改进提供了必要的深层理解。

Level	Focus	Key Process Area	PSP	TSP
5 Optimizing	Continuous Process Improvement	Defect Prevention	X	X
		Technology Change Management	X	X
		Process Change Management	X	X
4 Managed	Product and Process Quality	Quantitative Process Management	X	X
		Software Quality Management		
3 Defined	Engineering Process	Organization Process Focus	X	X
		Organization Process Definition	X	X
		Training Program		
		Integrated Software Management	X	X
		Software Product Engineering	X	X
		Intergroup Coordination		X
		Peer Reviews	X	X
2 Repeatable	Project Management	Requirements Management	X	X
		Software Project Planning	X	X
		Software Project Tracking	X	X
		Software Quality Assurance		X
		Software Configuration Management		X
		Software Subcontract Management		

Table 3: PSP and TSP coverage of CMM key process areas

CMMI模型是为美国国防部(DOD)鉴别用途的软件分包商所开发的。但对于DOD以外的其他组织来说，它也为组织的过程改进提供了一个有用的框架和强大的促进作用。许多组织的经验都表明CMMI对于软件组织改进它们的绩效是非常有效的。

一旦组织开始过程改进，并向CMMI2级发展，PSP向工程师们展示了如何以专业的方式来完成他们的任务。尽管是一个相对较新的概念，PSP已经经实践证明了它是如何帮助工程师改进计划、工作跟踪以及生产高品质产品的能力的。

工程团队接受了PSP培训后，他们需要将这一先进的流程方法运用到项目中。TSP能够指导这些团队启动他们的项目，计划并管理他们的工作。也许最重要的方面在于，PSP指导管理者如何引领并指导他们的软件队伍始终如一地发挥他们的最佳状态。