**Методология TCP(Team software process)**

**История**

В 1996 году Уоттс Хамфри разработал первоначальную версию процесса TSP. Его целью было обеспечить оперативный процесс, который помог бы инженерам последовательно выполнять качественную работу. Он сконструировал исходный TSP0-процесс как можно более простым, попробовал его с двумя командами, а затем проанализировал результаты, чтобы увидеть, как он работает. Затем он определил, в каких группах необходимы дальнейшие указания, и усовершенствовал процесс, обеспечивающий это руководство. Первый процесс TSP0 был разработан для групп, прошедших подготовку в PSP, которые не прошли никакой подготовки или инструкций, кроме тех, которые были предоставлены процессом TSP и непосредственным управлением команды.

Основываясь на результатах двух первоначальных групп TSP, было ясно, что TSP помог инженерам выполнять дисциплинированную работу, но что необходимы дополнительные рекомендации и поддержка. Было также очевидно, что руководство должно в целом поддерживать процесс ПТС. Дополнительный процесс TSP0.1 затем использовался дополнительными командами, предоставляя дополнительную информацию о необходимых усовершенствованиях процесса.

В течение следующих трех лет Хамфри разработал еще девять версий TSP. Вначале его задача состояла в том, чтобы выяснить, может ли командный процесс общего назначения помочь инженерным командам выполнять свою работу. После того, как стало ясно, что TCP выполнил эту основную задачу, его усилия по развитию процесса TCP были направлены на упрощение процесса, сокращение его размера и обеспечение поддержки и руководства, необходимых для того, чтобы сделать этот процесс наиболее эффективным и полезным. В результате последние версии TSP значительно меньше, чем версии TSP0.1 и TSP0.2, разработанные в конце 1996 и начале 1997 года.

Поскольку все больше групп использовали процесс TSP, были разработаны несколько методов внедрения, чтобы помочь инженерам и менеджерам создавать группы, лучше следить за процессом и периодически переоценивать и перепланировать проекты. Различные прототипные инструменты поддержки также были разработаны для упрощения планирования, записи данных, анализа данных и отчетности по проектам инженеров.

**Командная работа**

Команды необходимы для большинства разработческих проектов. Хотя некоторые небольшие аппаратные или программные продукты могут разрабатываться отдельными людьми, масштаб и сложность современных систем таковы, а спрос на короткие графики настолько велик, что больше не практично для одного человека выполнять большинство инженерных работ. Разработка систем - это командная деятельность, и эффективность команды во многом определяет качество техники.

Есть разные виды команд. Например, в спорте позиции баскетбольной команды динамичны, в то время как члены бейсбольной команды имеют более статические роли. Однако в обоих случаях члены должны работать совместно. И наоборот, команды борьбы и трека состоят из отдельных конкурентов, которые не взаимодействуют динамически, хотя члены поддерживают друг друга социально и эмоционально.

Команды разработчиков часто ведут себя как бейсбольные или баскетбольные команды. Несмотря на то, что они могут иметь несколько специальностей, все члены работают для достижения одной цели. Тем не менее, в командах по техническому обслуживанию и совершенствованию систем инженеры часто работают относительно независимо друг от друга, подобно командам борьбы и трека.

Команда - это больше, чем просто группа людей, которые работают вместе. Командная работа требует практики и требует специальных навыков. Команды требуют общих процессов; Им нужны согласованные цели; И им необходимо эффективное руководство и руководство. Методы руководства и руководства такими командами хорошо известны, но они не очевидны. Институт инженерии программного обеспечения (SEI) поддерживает TSP как средство для руководства инженерами и их менеджерами в использовании эффективных методов совместной работы.

**Эффективные команды**

Чтобы быть эффективными, команды должны обладать надлежащей квалификацией и иметь возможность работать в качестве единых подразделений. Эффективные команды имеют определенные общие характеристики:

* Члены опытные.
* Цель команды важна, определена, видима и реалистична.
* Ресурсы команды соответствуют задаче.
* Члены мотивированы и привержены достижению цели команды.
* Члены сотрудничают и поддерживают друг друга.
* Члены дисциплины в своей работе.

TSP предназначен для определения условий, характеризующих эффективные команды. Принципы тимбилдинга, используемые в TSP для установления этих условий, таковы:

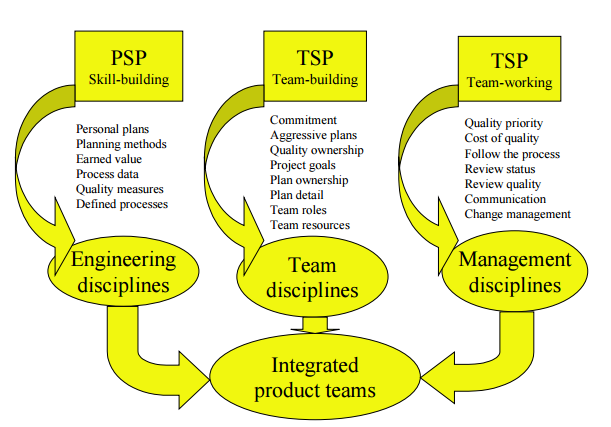
* Члены группы устанавливают общие цели и определенные роли.
* Команда разрабатывает согласованную стратегию.
* Члены команды определяют общий процесс их работы.
* Все члены команды участвуют в составлении плана, и каждый член группы знает свою личную роль в этом плане.
* Команда согласовывает план с руководством.
* Руководство анализирует и принимает согласованный план.
* Члены команды выполняют работу так, как они планировали это сделать.
* Члены команды общаются свободно и часто.
* Команда формирует сплоченную группу: члены сотрудничают, и все они привержены достижению цели.
* Инженеры знают свой статус, получают отклики на свою работу и руководство, которые поддерживают их мотивацию.

Для эффективного формирования команды требуется, чтобы члены действительно понимали, что они должны делать, соглашаться, как выполнять работу, и полагать, что их план достижим. Все эти условия могут быть установлены путем привлечения инженеров к разработке собственных планов. Тогда, предполагая, что эти планы грамотно составлены, команды могут почти всегда продавать свои планы руководству

Хотя все эти условия необходимы для эффективной совместной работы, конкретные способы установления этих условий не очевидны. TSP дает четкие указания, что организации должны создавать эффективные инженерные группы.

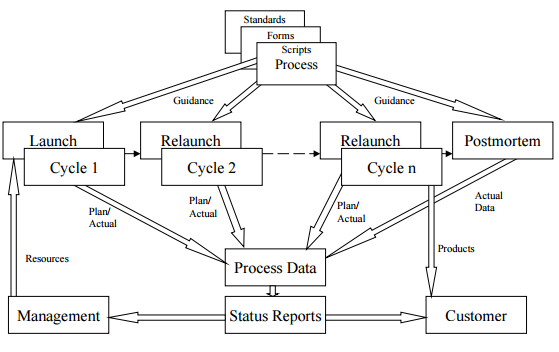
**Структура TSP**

Основные элементы процесса TSP показаны на рисунке ниже. Прежде чем участники смогут участвовать в команде TSP, они должны знать, как выполнять дисциплинированную работу. Как показано на этом рисунке, обучение в процессе персонального программного обеспечения (PSP) необходимо для обеспечения инженеров знаниями и навыками использования TSP. Обучение PSP включает изучение того, как составлять подробные планы, собирать и использовать данные процесса, разрабатывать планы заработанных стоимостей, использовать заработанную стоимость для отслеживания проекта, измерять и контролировать качество продукции, а также определять и использовать операционные процессы. Инженеры должны быть обучены этим навыкам, прежде чем они смогут участвовать в построении команды TSP или следовать установленному процессу TSP.



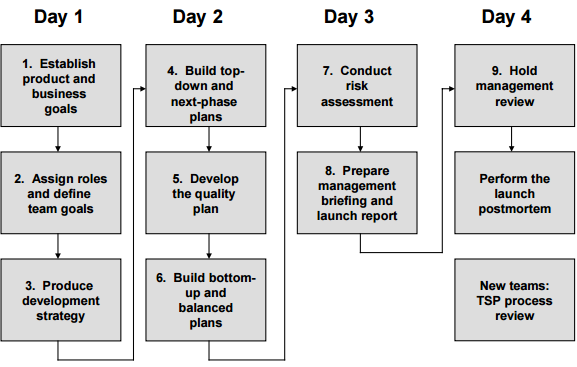
Хотя существует множество способов создания команд, все они требуют, чтобы отдельные лица работали вместе над выполнением некоторых сложных задач. В TSP эта сложная задача создания команды - это четырехдневный процесс планирования, который называется запуском команды. В начале все члены команды разрабатывают стратегию, процесс и план выполнения своего проекта. После завершения запуска команда следует своему определенному процессу, чтобы выполнить эту работу.

Как показано на рисунке ниже, команды TSP периодически обновляются. Поскольку процесс TSP следует итеративной и развивающейся стратегии развития, периодические перезапуска необходимы, чтобы каждый этап или цикл можно было планировать на основе знаний, полученных в предыдущем цикле. Перезапуск также необходим для обновления подробных планов инженеров, которые обычно точны всего несколько месяцев. Причина повторного запуска заключается в том, что подробные планы могут быть точными только в течение нескольких месяцев. В запуске TSP команды составляют общий план и подробный план на следующие три-четыре месяца. После того, как члены команды завершат всю или большую часть следующего этапа или цикла проекта, при необходимости они пересматривают общий план и составляют новый подробный план, охватывающий следующие три-четыре месяца. Они руководствуются этим в процессе перезапуска TSP.



**Запуск команды TSP**

После того, как члены команды прошли надлежащую подготовку и сформирована команда, вся команда участвует в запуске команды TSP. Процесс запуска показан в сценарии запуска на рисунке ниже. На каждом из 9 собраний по запуску есть сценарий, который описывает действия достаточно подробно, чтобы обученный тренер запуска мог вести команду через необходимые шаги. Следуя за процессом запуска, команды составляют подробный план. Чтобы стать сплоченным и эффективным рабочим подразделением, все члены команды должны быть привержены плану. Чтобы построить это обязательство, TSP привлекает всех членов команды к разработке этого плана. Таким образом, завершив процесс запуска TSP, все члены команды будут участвовать в составлении плана, и все они согласятся и будут привержены плану, который они подготовили.



**Процесс командной работы TSP**

Как только команда TSP будет запущена, основная задача - обеспечить, чтобы все члены команды следовали плану. Это включает в себя следующие основные темы:

* Руководство командой
* Процессуальная дисциплина
* Отслеживание ошибок
* Связь
* Управленческая отчетность;
* Поддержание плана
* Оценка завершения проекта
* Перебалансировка рабочей нагрузки
* Возобновление проекта
* Управление качеством TSP

**Управление качеством TSP**

Хотя большинство организаций согласны с тем, что качество важно, лишь немногие команды знают, как управлять качеством своей продукции. Кроме того, отсутствуют общие методы предотвращения впрыска дефектов. Люди разрабатывают программное обеспечение, и люди делают ошибки. Эти ошибки являются источником дефектов программного обеспечения. В TSP основное внимание уделяется контролю дефектов.

Чтобы управлять качеством, команды должны принимать меры по обеспечению качества, устанавливать цели в области качества, разрабатывать планы для достижения этих целей, оценивать прогресс по сравнению с планами и предпринимать меры по исправлению ситуации, когда цели не выполняются. TSP показывает командам, как это сделать. Элементы управления качеством TSP составляют план качества, идентифицируют проблемы качества, а также находят и предотвращают проблемы с качеством. Эти темы рассматриваются в следующих параграфах.

**План качества**

Во время процесса команды TSP составляют план качества. Основываясь на оценочном размере продукта и исторических данных о коэффициентах впрыска дефектов, они оценивают, сколько дефектов они будут вводить в каждой фазе. В тех случаях, когда команды не имеют исторических данных о вводе дефектов, они могут использовать руководящие принципы планирования качества TSP, приведенные в таблице 3. Это поможет им определить цели качества. Как только инженеры оценят дефекты, которые будут введены, они оценивают удаление дефектов, снова используя исторические данные или рекомендации по качеству TSP. Эти оценки удаления основываются на урожае для каждой фазы удаления дефектов. Здесь доходность относится к проценту дефектов в продукте при вводе фазы, которые удаляются в этой фазе. После того, как были сделаны оценки впрыска и удаления, команда может разработать план качества. Наконец, команда изучает план качества, чтобы определить, являются ли параметры качества разумными и соответствуют ли они целям качества команды. Если нет, инженеры корректируют оценки и создают новый план качества.

**Внедрение TCP**

TSP внедряется в индустриальную и академическую среду. Вводный процесс и учебник TSPi доступны для обучения в командах университетских команд [Humphrey 00]. Прежде чем начать курс TSPi, студенты должны пройти курс PSP с текстом «Дисциплина для разработки программного обеспечения» или текстом «Введение к личному процессу программного обеспечения» [Humphrey 95, Humphrey 97]. Курс TSPi преподается в ряде университетов, и ранний опыт показывает, что студенты и преподаватели считают его полезным. Тем не менее, пока не опубликованы исследования по использованию процесса и методов TSPi.

Внедрение TSP в инженерные организации является основным направлением усилий TSP в Институте разработки программного обеспечения (SEI). TSP был разработан для инженерных команд, и его внедрение первоначально предназначалось для команд, разрабатывающих программные продукты. Для поддержки использования промышленными группами, которые включают в себя помимо программных специальностей, SEI разработала вводный курс PSP для профессионалов, которые не владеют программным обеспечением. Он также ввел серию учебных и квалификационных программ, с тем чтобы организации могли получить своих собственных инструкторов PSP. Кроме того, SEI обеспечивает подготовку тренеров TSP, чтобы организации могли запускать и тренировать собственные команды TSP. SEI установил отношения с несколькими партнерами по переходу, которые имеют квалификацию для обучения PSP и для подготовки команд TSP. (Для получения дополнительной информации по этим темам см. [Www.sei.cmu.edu/tsp](http://Www.sei.cmu.edu/tsp).)

**Опыт работы с TSP**

Хотя TSP является относительно новым, о результатах сообщается рядом организаций. Некоторые примеры обобщены в следующих параграфах.

* Teradyne обнаружил, что до TSP уровни дефектов в интеграционном тестировании, системном тестировании, полевых испытаниях и использовании клиентов составляли в среднем около 20 дефектов на KLOC (1000 строк кода или LOC). Первый проект TSP снизил эти уровни до 1 дефекта на KLOC. Так как затраты на поиск и исправление каждого дефекта в среднем составляют 12 инженерных часов, Теридайн сэкономил 228 инженерных часов на каждую 1000 разработанных программ. Так как типичная стоимость кода и единицы измерения 1000 LOC составляет около 50 инженерных часов, экономия затрат на ремонт дефектов была примерно в 4,5 раза выше затрат на производство программ в первую очередь.
* База ВВС Хилл, недалеко от Солт-Лейк-Сити, штат Юта, является первой организацией в штате США, которая будет оценена на уровне 5 CMM [Paulk 95]. Первый проект TSP в Hill показал, что производительность команды улучшилась на 123%, а время тестирования сократилось с организационной средней в 22% до 2,7% от расписания проекта [Webb 00].
* Boeing, на большом авионическом проекте, показал результаты, показанные на рисунках 8 и 9. Сокращение времени тестирования системы на 94% привело к существенному улучшению графика проекта и позволило Boeing досрочно доставить высококачественный продукт.