


Презентация

Практическая необходимость.
Документация. Стандарты. СММ.



Практическая необходимость

Описывая практичность необходимо описать:

- Интерфейс пользователя. Должен быть назван тип интерфейса: строка команд, меню, окно, функциональные клавиши и др.
- Требуемые знания. Должны быть определены конкретные знания, которые необходимо осветить пользователю для применения соответственного продукта. Например:
 - Знания соответственной технической области
 - Знания ОС
 - Знания, полученные в результате специального обучения.
- Адаптация к потребностям пользователя. Если продукт может переделываться (адаптироваться) пользователем, то должны быть указаны инструментальные средства для проведения такой настройки и условия их применения. Например: изменение параметров; изменение алгоритма вычислений и назначение функциональных клавиш.
- Защита от нарушения авторских прав. В описании продукта должны быть указаны виды и средства защиты: технической защиты от потребителя; шифрование даты окончания использования продукта, напоминание о плате за копию.
- Эффективность применения и удовлетворение потребностям пользователя. В описании продукта может быть вписана информация по эффективности применения продукта. Описывая эффективность, необходимо описать информацию о характере поведения продукта во времени.

Документация пользователя

Это полный список документов, поставляемый в печатном или другом виде, который обеспечивает применение продукта, а так же является его неотъемлемой частью. Документация пользователя должна отвечать следующим положениям:

- **Полнота.** Документация пользователя должна содержать информацию необходимую для использования продукта; в ней должно быть полное описание всех функций, установленных в описании продукта и все вызываемые пользователем из программы. Кроме того, граничные значения, заданные в описании продукта должны быть продублированы в документации пользователя. Если установка (инициализация) продукта может быть проведена пользователем, то в документации пользователя должно быть включено руководство по установке продукта, содержащее всю необходимую информацию; если сопровождение продукта может проводиться пользователем, то в документации пользователя должно быть включено руководство по оптимизации программы, содержащая всю информацию, которая необходимо для обеспечения данного вида содержания.
- **Правильность.** Вся информация в документации пользователя должна быть правильной, кроме того предоставленные данные в информации не должны содержать ошибок.
- **Непротиворечивость.** Документы не должны противоречить сами себе друг другу.
- **Понятность.** Документация пользователя должна быть понятной для сообщества пользователей выполняющих указанную задачу. Например, по средствам использующихся в ней соответственным образом терминов.
- **Простота обозрения.** Документация пользователя должна быть достаточно просто для изучения пользователя.

Программные данные

Должны соответствовать всем обязательным формулировкам, приведенным в описании продукта и документации пользователя.

- Надежность. Это требование должно удовлетворять в случаях, когда возможность реализации при используемых ограничениях; имеют место реализации возможностей вне заданных ограничений; неправильные исходные данные вводятся только пользователем или другими программами; нарушается инструкции заданные в документации пользователя.
- Практичность.
- Понятность. Сообщения и результаты выполнения программы должны быть понятны. Например: путем соответственного выбора терминов; путем представления исходной информации путем пояснения функций подсказки. В сообщения об ошибках должны сообщать информацию, разъясняющую причину или способ исправления соответствующих ошибок из-за неправильности изменения продукта. Например, путем ссылки на элемент документации пользователя.
- Простота обозрения. Программа должна предъявлять пользователю информацию в качественном виде, чтобы данные легко было воспринимать и читать.
- Простота использования. Исполнение функции приводящим к серьезным последствиям при эксплуатации системы должно быть обратимо или программа должна выводить предупреждение о последствиях выполнения данной функции и запрашивать подтверждение перед выполнением команды.

Стандарты

В настоящий момент существует несколько стандартов связанных с оценкой качества программных средств, которые обеспечивает организация разработчик. К наиболее известным относятся **ISO-9000** (ISO-9000-9004), **CMM** (**C**apability **M**aturity **M**odel) т.е. модель зрелости (совершенствования) процессов создания ПО , предложенная **SEI** (**S**oftware **E**ngineering **I**nstitute) – институтом технологии программирования при Университете Карнеги-Меллона.

Рабочая версия международного стандарта ISO/IEC 15504. Information technology - Software Process Assessment. С другой стороны эта версия более известна под названием **SPICE** (**S**oftware **P**rocess **I**mprovement and **C**apability **dE**termination) - определение возможностей и улучшение процесса создания программного обеспечения.

В серии стандартов ISO-9000 сформулированы только необходимые условия для достижения некоторого минимального уровня организованности процесса, но не дается никаких рекомендаций по дальнейшему совершенствованию процессов.

Capability Maturity Model (CMM)

Представляет собой совокупность критериев оценки зрелости организации разработчика и рецептов улучшения соответствующих процессов.

Примечание: Изначально CMM разрабатывался и развивался как методика, позволяющая крупным правительственным организациям США выбирать наилучших поставщиков ПО. Для этого предполагалось создать исчерпывающее описание способов оценки процессов разработки ПО и методики их дальнейшего усовершенствования. В итоге, авторы смогли добиться такой степени детализации, что стандарт оказался пригодным и для обычных компаний-разработчиков, желающих улучшить существующие процессы разработки и привести их к определенным стандартам.

CMM определяет пять уровней зрелости организаций-разработчиков, причем каждый следующий уровень включает в себя все ключевые характеристики предыдущих.

Начальный уровень (initial level)

- Описан в стандарте в качестве основы для сравнения со следующими уровнями. На предприятии такого уровня организации не существует стабильных условий для созданий качественного программного обеспечения. Результат любого проекта целиком и полностью зависит от личных качеств менеджера и опыта программистов, причем успех в одном проекте может быть повторен только в случае назначения тех же менеджеров и программистов на следующий проект. Более того, если такие менеджеры или программисты уходят с предприятия, то резко снижается качество программных продуктов. В стрессовых ситуациях процесс разработки сводится к написанию кода и его минимальному тестированию.

Повторяемый уровень (repeatable level)

На предприятии ввести технологии управления проектами. При этом планирование и управление проектами основывается на накопленном опыте, существуют стандарты на разрабатываемое программное обеспечение (причем обеспечивается следование этим стандартам). Создается специальная группа обеспечения качества. В случае необходимости организация может взаимодействовать с субподрядчиками. В критических условиях процесс имеет тенденцию скатываться на начальный уровень.

Определенный уровень (defined level)

Характеризуется тем, что стандартный процесс создания и сопровождения программного обеспечения полностью документирован (включая и разработку ПО, и управление проектами). Подразумевается, что в процессе стандартизации происходит переход на наиболее эффективные практики и технологии. Для создания и поддержания подобного стандарта в организации должна быть создана специальная группа. Наконец, обязательным условием для достижения данного уровня является наличие на предприятии программы постоянного повышения квалификации и обучения сотрудников. Начиная с этого уровня, организация перестает зависеть от качеств конкретных разработчиков, и не имеет тенденции скатываться на уровень ниже в стрессовых ситуациях.

Управляемый уровень (managed level)

В организации устанавливаются количественные показатели качества – как на программные продукты, так и на процесс в целом. Таким образом, более совершенное управление проектами достигается за счет уменьшения отклонений различных показателей проекта. При этом осмысленные вариации в производительности процесса можно отличить от случайных вариаций (шума), особенно в хорошо освоенных областях.

Оптимизирующий уровень (optimizing level)

Характеризуется тем, что мероприятия по улучшению применяются не только к существующим процессам, но и для оценки эффективности ввода новых технологий. Основной задачей всей организации на этом уровне является постоянное улучшение существующих процессов. При этом улучшение процессов в идеале должно помогать предупреждать возможные ошибки или дефекты. Должны вестись работы по уменьшению стоимости разработки программного обеспечения, например, с помощью создания и повторного использования компонент.

Сертификационная оценка

Сертификационная оценка соответствия всех ключевых областей проводится по 10-балльной шкале. Для успешной квалификации данной ключевой области необходимо набрать не менее 6 баллов. Оценка ключевой области осуществляется по следующим показателям:

- Заинтересованность руководства в данной области (планируется ли практическое внедрение данной ключевой области, существует ли понимание у руководства необходимости данной области и т.д.).
- Насколько широко данная область применяется в организации (например, оценке в 4 балла соответствует фрагментарное применение).
- Использование данной области на практике (оценке в 0 баллов соответствует полное отсутствие какого-либо эффекта, а оценка в 8 баллов выставляется при наличии систематического и измеримого положительного результата практически во всей организации).