

ГОСТ Р ИСО/МЭК 12207-2010

Процессы жизненного цикла программных средств

Обозначения стандартов

Буква "Р" в обозначении ГОСТ означает, что он действует только на территории РФ. "РД" - руководящие документы.

Первые цифры - это номер регистрации, а последние цифры - год принятия, то есть год регистрации данного стандарта.

ГОСТ 19.xxx - Единая система программной документации (ЕСПД).

ГОСТ 34.xxx - Информационная технология. Комплекс стандартов на автоматизированные системы.

ISO - Международная организация по стандартизации.

О стандарте

Данный стандарт устанавливает общую структуру процессов жизненного цикла программных средств, на которую можно ориентироваться в программной индустрии. Он определяет процессы, виды деятельности и задачи, которые используются при приобретении программного продукта или услуги, а также при поставке, разработке, применении, сопровождении и прекращении применения программных продуктов.

Настоящий стандарт используется при приобретении систем, программных продуктов и услуг, при их поставке, разработке, применении по назначению, сопровождении и прекращении применения программных продуктов и программных компонентов системы как в самой организации, так и вне ее.

Термины

Аудит (audit): независимая оценка программных продуктов и процессов, проводимая уполномоченным лицом с целью оценить их соответствие требованиям.

Базовая линия (baseline): спецификация или продукт, которые были официально рассмотрены и согласованы, чтобы служить основой для дальнейшего развития (могут быть изменены только посредством официальных и контролируемых процедур изменения).

Составная часть конфигурации (configuration item): объект в пределах конфигурации, который удовлетворяет некоторой функции целевого применения и может быть однозначно идентифицирован в данный момент времени.

Обеспечивающая система (enabling system): система, которая служит дополнением к рассматриваемой системе на протяжении стадий ее жизненного цикла, но не обязательно вносит непосредственный вклад в ее функционирование.

Жизненный цикл (life cycle): развитие системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии разработки концепции и заканчивая прекращением применения.

Непоставляемая составная часть (non-deliverable item): техническое средство или программный продукт, который не требуется поставлять по условиям контракта, но который может использоваться в разработке программного продукта.

Тестовое покрытие (test coverage): степень, с которой данный тест проверяет требования для системы или программного продукта.

Валидация (validation): подтверждение на основе представления объективных свидетельств того, что выполнены требования, предназначенные для конкретного использования или применения.

Применение стандарта

Программное средство рассматривается как единая часть общей системы, выполняющая определенные функции, что осуществляется посредством выделения требований к программным средствам из требований к системе, проектирования, производства программных средств и объединения их в систему.

Организация — это группа лиц с определенными обязанностями и полномочиями, объединенных для реализации конкретных целей. Если организация полностью или частично входит в контрактное соглашение (договор), то это — сторона.

Организация или сторона получают свои наименования от процессов, за которые они ответственны. Например, организация называется приобретающей стороной, если она выполняет процесс приобретения. Таким образом, следующие термины указывают на организацию или сторону, ответственную за выполнение процесса со сходным названием: приобретающая сторона, поставщик, исполнитель, сопровождающая сторона и оператор.

Процессы и организации (стороны) связаны только функционально. Можно отметить, что и организация может выполнять один или несколько процессов, и процесс может реализовываться одной или несколькими организациями.

Жизненный цикл начинается от замысла или потребности, которая может быть удовлетворена полностью или частично программным средством, и завершается прекращением применения этого программного средства. Такая архитектура создается совокупностью процессов и взаимосвязями между ними.

Принципы определения процессов жизненного цикла:

- связности (процессы являются связными и соединяются оптимальным образом, считающимся практичным и выполнимым);
- ответственности (процесс передается под ответственность какой-либо организации или стороне в пределах жизненного цикла программного средства).

Каждый процесс можно описать с помощью нескольких атрибутов:

- ❶ название (передает область применения процесса как целого);
- ❷ цель (описывает конечные цели выполнения процесса);
- ❸ выходы (наблюдаемые результаты, ожидаемые при успешном выполнении процесса);
- ❹ деятельность (список действий, используемых для достижения выходов);
- ❺ задачи (требования, рекомендации или допустимые действия, предназначенные для поддержки достижения выходов процесса).

Жизненный цикл

Модель жизненного цикла представляется в виде последовательности стадий, которые могут перекрываться и (или) повторяться циклически в соответствии с областью применения, размером, сложностью, потребностью в изменениях и возможностях. Каждая стадия описывается формулировкой цели и выходов. Процессы и действия жизненного цикла отбираются и исполняются на этих стадиях для полного удовлетворения цели и результатам каждой стадии.

Использовать какую-либо конкретную модель жизненного цикла не требуется. Однако необходимо, чтобы в каждом проекте определялась подходящая модель жизненного цикла, предпочтительно та, которая уже выбиралась организацией для применения в различных проектах. Применение модели жизненного цикла обеспечивает средства для установления зависимой от времени последовательности, необходимой для менеджмента проекта.

ЖИЗНЕННЫЙ ЦИКЛ ПО

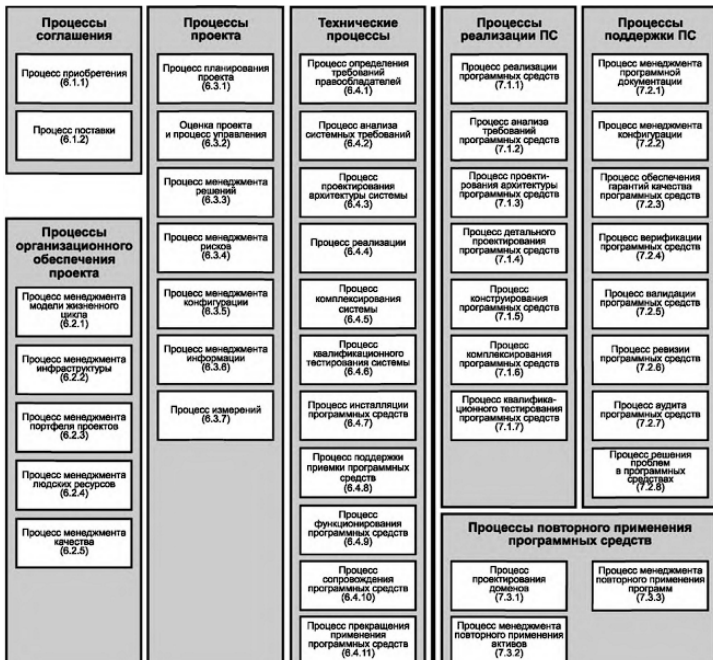


Примеры моделей жизненного цикла

- **Каскадная (Waterfall Model):** последовательное выполнение этапов разработки, где каждый следующий начинается только после завершения предыдущего.
- **V-образная (V-Model):** расширяет каскадную, вводя этапы тестирования для каждого этапа разработки.
- **Инкрементная:** разработка осуществляется поэтапно, с последовательным добавлением функционала. Каждый инкремент представляет собой работоспособную версию продукта с новым набором функций.

Примеры моделей жизненного цикла

- **Итерационная:** процесс разработки разбивается на повторяющиеся циклы (итерации), в ходе которых продукт постепенно дорабатывается и улучшается. Каждая итерация включает полный цикл разработки: от анализа требований до тестирования.
- **Спиральная:** сочетает элементы каскадной и итерационной моделей, акцентируя внимание на управлении рисками.
- **Agile-модель:** гибкая методология, ориентированная на быструю доставку работающего продукта и тесное взаимодействие с заказчиком.



Процессы в контексте системы

Процессы соглашения определяют действия, необходимые для выработки соглашений между двумя организациями. Если реализуется *процесс приобретения*, то он обеспечивает средства для проведения деловой деятельности с поставщиком продуктов, предоставляемых для применения в функционирующей системе, услугах поддержки этой системы или элементах системы, разработанных в рамках проекта. Если реализуется *процесс поставки*, то он обеспечивает средства для проведения проекта, в котором результатом является продукт или услуга, поставляемые приобретающей стороне.

Процессы организационного обеспечения проекта осуществляют менеджмент возможностей организаций приобретать и поставлять продукты или услуги через инициализацию, поддержку и управление проектами. Эти процессы обеспечивают ресурсы и инфраструктуру, необходимые для поддержки проектов, и гарантируют удовлетворение организационных целей и установленных соглашений.

Существуют две категории **процессов проекта**: *процессы менеджмента* (планирование, выполнение, оценка и управление продвижением проекта) и *процессы поддержки проекта* (обеспечивают выполнение специализированных целей менеджмента).

Отдельные процессы менеджмента проекта могут привлекаться в любое время жизненного цикла и на любом уровне иерархии проекта в соответствии с планами проекта или возникновением непредвиденных событий.

Технические процессы используются для определения требований к системе, преобразования требований в полезный продукт, для разрешения постоянного копирования продукта (где это необходимо), применения продукта, обеспечения требуемых услуг, поддержания обеспечения этих услуг и изъятия продукта из обращения, если он не используется при оказании услуги.

Специальные процессы программных средств

Процессы реализации программных средств используются для создания конкретного элемента системы (составной части), выполненного в виде программного средства. Эти процессы преобразуют заданные характеристики поведения, интерфейсы и ограничения на реализацию в действия, результатом которых становится системный элемент, удовлетворяющий требованиям, вытекающим из системных требований.

Процессы поддержки программных средств предусматривают специально сфокусированную совокупность действий, направленных на выполнение специализированного программного процесса. Любой поддерживающий процесс помогает процессу реализации программных средств как единое целое с обособленной целью, внося вклад в успех и качество программного проекта.

Группа процессов повторного применения программных средств состоит из трех процессов (проектирования доменов, менеджмента повторного применения активов и программ), которые поддерживают возможности организации использовать повторно составные части программных средств за границами проекта. Эти процессы уникальны, поскольку, в соответствии с их природой, они используются вне границ какого-либо конкретного проекта.

