

Жизненный цикл программного средства.

Под **жизненным циклом ПС** (software life cycle) понимают весь период его разработки и эксплуатации (использования), начиная от момента возникновения замысла ПС и кончая прекращением всех видов его использования. Жизненный цикл охватывает довольно сложный процесс создания и использования ПС (software process). Этот процесс может быть организован по-разному для разных классов ПС и в зависимости от особенностей коллектива разработчиков.

Основным стандартом, характеризующим жизненный цикл ПС сейчас является:

- **Гибкая разработка ПС** (Agile software development), **agile-методы** — обобщающий термин для целого ряда подходов и практик, основанных на ценностях Манифеста гибкой разработки программного обеспечения.

К гибкой разработке, в частности, относят экстремальное программирование, DSDM, Scrum, FDD, BDD и др.

Применяется как эффективная практика организации труда небольших групп (которые делают однородную творческую работу) в объединении с управлением ими комбинированным (либеральным и демократическим) методом.

Также в настоящее время можно выделить 5 основных подходов к организации процесса создания и использования ПС:

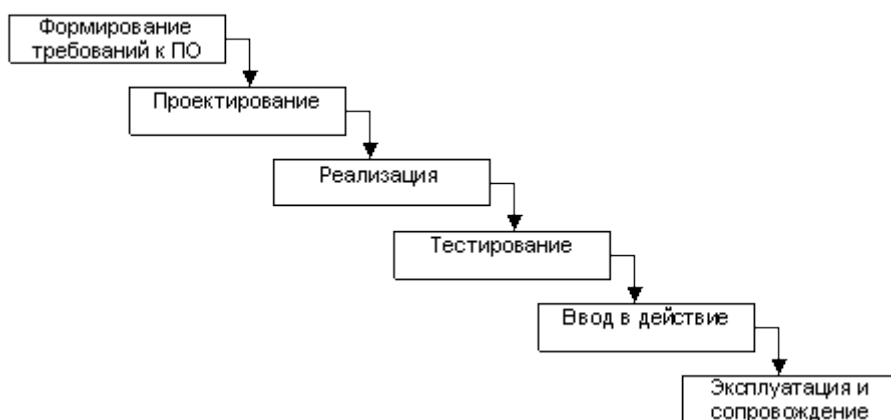
- **Водопадный подход.** При таком подходе разработка ПС состоит из цепочки этапов. На каждом этапе создаются документы, используемые на последующем этапе. В исходном документе фиксируются требования к ПС. В конце этой цепочки создаются программы, включаемые в ПС.
- **Исследовательское программирование.** Этот подход предполагает быструю (насколько это возможно) реализацию рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции. После экспериментального применения реализованных программ производится их модификация с целью сделать их более полезными для пользователей. Этот процесс повторяется до тех пор, пока ПС не будет достаточно приемлемо для пользователей. Такой подход применялся на ранних этапах развития программирования, когда технологии программирования не придавали большого значения (использовалась интуитивная технология). В настоящее время этот подход применяется для разработки таких ПС, для которых пользователи не могут точно сформулировать требования (например, для разработки систем искусственного интеллекта).
- **Прототипирование.** Этот подход моделирует начальную фазу исследовательского программирования вплоть до создания рабочих версий программ, предназначенных для проведения экспериментов с целью установить требования к ПС. В дальнейшем должна последовать разработка ПС по установленным требованиям в рамках какого-либо другого подхода (например, водопадного).
- **Формальные преобразования.** Этот подход включает разработку формальных спецификаций ПС и превращение их в программы путем корректных преобразований. На этом подходе базируется компьютерная технология (CASE-технология) разработки ПС.
- **Сборочное программирование.** Этот подход предполагает, что ПС конструируется, главным образом, из компонент, которые уже существуют. Должно быть некоторое хранилище (библиотека) таких компонент, каждая из которых может многократно использоваться в разных ПС. Такие компоненты называются повторно используемыми (reusable). Процесс разработки ПС при данном подходе состоит скорее из сборки программ из компонент, чем из их программирования.

Модели жизненного цикла программных средств

Модель жизненного цикла - это схема выполнения работ и задач в рамках процессов, обеспечивающих разработку, эксплуатацию и сопровождение программного продукта, и отражающая эволюцию ПС, начиная от формулировки требований к ней до прекращения пользоваться ею.

К настоящему времени наибольшее распространение получили следующие основные модели ЖЦ:

- **Каскадная модель.** Её основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем. Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков. Каскадная схема разработки программного средства:



- **Спиральная модель.** Делает упор на анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПС, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации. Схема спиральной модели жизненного цикла:

