**Модели жизненного цикла программных средств**

***Модель жизненного цикла:***структура, состоящая из процес­сов, работ и задач, включающих в себя разработку, эксплуата­цию и сопровождение программного продукта, охватывающая жизнь системы от установления требований к ней до прекраще­ния ее использования.

**К настоящему времени наибольшее распространение получи­ли следующие основные модели ЖЦ:**

•  каскадная модель (70-80-е годы 20 века);

•  спиральная модель (80-90-е годы 20 века).

В изначально существовавших однородных ИС каждое при­ложение представляло собой единое целое. Для разработки тако­го типа приложений применялся каскадный способ. Его 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 (рис. 1). Каждый этап завершается выпуском полно­го комплекта документации, достаточной для того, чтобы разра­ботка могла быть продолжена другой командой разработчиков.

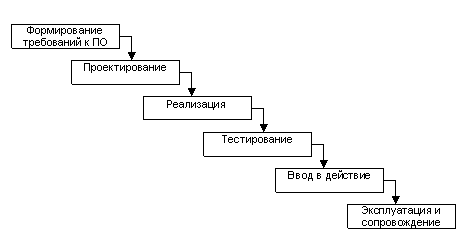


Рис. 1. Каскадная схема разработки программного средства

Положительные стороны применения каскадного подхода зак­лючаются в следующем:

•  на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласован­ности;

•  выполняемые в логичной последовательности этапы работ по­зволяют планировать сроки завершения всех работ и соответ­ствующие затраты.

Каскадный подход хорошо зарекомендовал себя при постро­ении ИС, для которых в самом начале разработки можно доста­точно точно и полно сформулировать все требования, с тем что­бы предоставить разработчикам свободу реализовать их как мож­но лучше с технической точки зрения. В эту категорию попадают сложные расчетные системы, системы реального времени и дру­гие подобные задачи.

Основным недостатком каскадного подхода является существен­ное запаздывание с получением результатов. Согласование резуль­татов с пользователями производится только в точках, планируе­мых после завершения каждого этапа работ, требования к ИС «за­морожены» в виде технического задания на все время ее создания. Таким образом, пользователи могут внести свои замечания толь­ко после того, как работа над системой будет полностью заверше­на. В случае неточного изложения требований или их изменения в течение длительного периода создания ПС пользователи получа­ют систему, не удовлетворяющую их потребностям. Модели (как функциональные, так и информационные) автоматизируемого объекта могут устареть одновременно с их утверждением.

Для преодоления перечисленных проблем была предложена **спиральная модель** ЖЦ (рис. 2), делающая упор на начальные эпапы ЖЦ: анализ и проектирование. На этих этапах реализуе­мость технических решений проверяется путем создания прото­типов. Каждый виток спирали соответствует созданию фрагмен­та на нем уточняются цели и характеристики про­екта, определяется его качество и планируются работы следующего витка спирали. 

Рис. 2. Схема спиральной модели жизненного цикла

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. При итера­тивном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача — как можно быстрее показать пользователям системы работоспособ­ный продукт, тем самым, активизируя процесс уточнения и до­полнения требований.

Основная проблема спирального цикла — определение мо­мента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла.