**Модель жизненного цикла**

Выбор модели жизненного цикла разработки и сопровождения СКЗИ «Dcrypt 1.0 v.2» обусловлен следующими факторами:

* изменяющиеся во времени требования к среде функционирования;
* обнаружение возможности улучшения функционирования СКЗИ «Dcrypt 1.0 v.2»;
* обнаружение новых уязвимостей примененных при разработке решений;
* максимально быстрое получение версии продукта, готовой к эксплуатации;
* планируется длительная поддержка и выпуск продукции.

Для разработки СКЗИ «Dcrypt 1.0 v.2» была выбрана спиральная модель жизненного цикла (см. рисунок Рисунок 1). Каждая итерация (виток спирали) представляет собой законченный цикл разработки, приводящий к выпуску внутренней или внешней версии изделия (или подмножества конечного продукта), которое совершенствуется от итерации к итерации, чтобы стать законченной системой.



Рисунок 2 – Модель жизненного цикла СКЗИ «Dcrypt 1.0 v.2»

Версия, получающаяся по завершении каждой итерации, является законченным продуктом, готовым к производству. После завершения такого производства продукт проходит тематическое исследование результаты которого представляются в ФСБ России.

Таким образом, каждый виток спирали соответствует созданию фрагмента или версии программного изделия, на нем уточняются цели и характеристики проекта, определяется его качество, планируются работы на следующем витке спирали.

Итерационная разработка существенно упрощает внесение изменений в проект при изменении требований заказчика. Она обеспечивает большую гибкость в управлении проектом, давая возможность внесения тактических изменений в разрабатываемое изделие. Например, можно сократить сроки разработки за счет снижения функциональности системы или использовать в качестве составных частей системы продукцию сторонних фирм вместо собственных разработок. Это может быть актуальным в условиях конкурентной борьбы, когда необходимо противостоять продвижению изделия, предлагаемого конкурентами.

Итерационный подход упрощает повторное использование компонентов (реализует компонентный подход к программированию). Это обусловлено тем, что гораздо проще выявить (идентифицировать) общие части проекта, когда они уже частично разработаны, чем пытаться выделить их в самом начале проекта. Анализ проекта после проведения нескольких начальных итераций позволяет выявить общие многократно используемые компоненты, которые на последующих итерациях будут совершенствоваться.

Спиральная модель позволяет получить более надежную и устойчивую систему. Это связано с тем, что по мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации. Одновременно могут корректироваться критические параметры эффективности, что в случае каскадной модели доступно только перед внедрением системы.

Итерационный подход дает возможность совершенствовать процесс разработки – анализ, проводимый в конце каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации.

Распределение ролей:

* Менеджер проекта: Генеральный директор
* Интегратор (ответственный за конфигурационное управление разработкой программного обеспечения): Ведущий инженер-программист
* Интегратор (ответственный за конфигурационное управление разработкой проектной документации): Технический писатель
* Интегратор (ответственный за конфигурационное управление подбором аппаратных платформ): Специалист по тестированию и контролю качества
* Тестирование (ответственный за тестирование аппаратных платформ): Системный администратор, инженер
* Права доступа: Права доступа внутри группы разработчиков не ограничиваются, т.е. все имеют доступ ко всем подпроектам.