# **ЛАБОРАТОРНАЯ РАБОТА № 4.**

**Процессы жизненного цикла программных средств**

**Цель работы:** Изучить различия процессов жизненного цикла (ЖЦ) ПО с точки зрения различных международных и национальных стандартов. Определить целесообразную для выполнения проекта модель ЖЦ. Выполнить технологический процесс кодирования ПО по выполняемому проекту.

Жизненный цикл программного обеспечения - период от момента появления идеи создания некоторого программного обеспечения до момента завершения его поддержки фирмой-разработчиком или фирмой, выполнявшей сопровождение. В ходе жизненного цикла ПО проходит через анализ предметной области, сбор требований, проектирование, кодирование, тестирование, сопровождение и др.

Основные стадии ЖЦ программных средств согласно стандарту ГОСТ Р ИСО/МЭК 12207-2010.:

* Стадия определения требований к ПО
* Стадия создания документации
* Стадия проектирования
* Стадия реализации
* Стадия тестирования
* Стадия развертывания проекта и его реализация

Общепринятой структуры жизненного цикла ПО не существует, поскольку она существенно зависит от целей, для которых это ПО разрабатывается и от решаемых им задач. Тем не менее, часто определяют основные элементы структуры жизненного цикла в виде модели жизненного цикла ПО. Модель жизненного цикла ПО выделяет конкретные наборы видов деятельности (обычно разбиваемых на еще более мелкие активности), артефактов, ролей и их взаимосвязи, а также дает рекомендации по организации процесса в целом.

**Определение целесообразной для выполнения проекта модели ЖЦ.**

В целях определения наиболее подходящей модели ЖЦ для нашего проекта нами были рассмотрены следующие популярные модели ЖЦ:

#### Каскадная модель жизненного цикла программного обеспечения (водопад): структура «шаг за шагом» данной модели не подходит нашему проекту, так как в процессе разработки требования к нашему ПО могут меняться. При таких условиях в данной модели мы получим большие и ничем необоснованные потери по времени;

#### V модель (разработка через тестирование): данная модель унаследовала структуру от каскадной модели, соответственно при выборе данной модели для нашего проекта, мы столкнемся с теми же большими и ничем необоснованными потерями по времени. Данная модель нам также не подходит;

#### 

* Инкрементная модель жизненного цикла программного обеспечения: оптимальное использование данной модели подразумевает, что: требования к конечной системе заранее четко определены и понятны, а проект большой или очень большой. Наш проект не удовлетворяет этим требованиям, поэтому данная модель нам также не подходит;



#### Спиральная модель жизненного цикла программного обеспечения: данная модель резонна для сложных и дорогих проектов, например, разработки системы документооборота для банка, так как подразумевает, что каждый следующий шаг требует большего анализа для оценки последствий, чем программирование. Получаем, что данная модель является необоснованно ресурсоемкой для нашего проекта, поэтому она нам не подходит;

#### Big Bang Model (модель большого взрыва): оптимальное использование данной модели подразумевает сосредоточение всех возможных ресурсов на разработке программного обеспечения и кодировании, практически не планируя или вообще не планируя (Требования понимаются и реализуются по мере их поступления). В нашем проекте частично присутствует эта модель(в том, что каждый делает своё), но подразумевает наличие достаточного планирования, чтобы не использовать данную модель;

#### «RAD Model» (rapid application development model или быстрая разработка приложений): данная модель может использоваться только при наличии высококвалифицированных и узкоспециализированных архитекторов, и подразумевает выполнение проекта в короткие сроки. Так как на сегодняшний день мы не обладаем достаточными навыками, средств на найм профессионалов нет, и сроки не сильно поджимают, нас не устраивает эта модель

* Agile Model (гибкая методология разработки): Не подходит нашему проекту, т.к. там подразумевается каждый раз прохождение полного цикла разработки, а мы работаем больше постепенно, то есть возвращаемся после каждого этапа на этап выше или в самое начало.

# 