

Конспекты по программированию

2 семестр

ИКТ
2021 — 2022

Автор:
Даниил Швалов

Оглавление

1	Процесс создания ПО	3
2	Жизненный цикл программного продукта	3
3	Каскадная модель	4
4	Спиральная модель	4
5	Agile	5
6	Kanban	5
7	Scrum	6

1. Процесс создания ПО

Процесс создания ПО – совокупность мероприятий, целью которых является создание или модернизация ПО.

1. Анализ предметной области (постановка задачи)
2. Разработка проекта системы
 - (а) Создание модели, отражающей основные функциональные требования, предъявляемые к программе
 - (б) Выбор метода решения (построение мат. модели)
 - (с) Разработка алгоритма – последовательности действий по решению задачи
3. Реализация программы на языке программирования (кодирование)
4. Анализ полученных результатов (тестирование)
5. Внедрение и сопровождение

Этап анализа состоит в исследовании системных требований и проблемы. Различают:

- анализ требований — исследование требований к системе;
- объектный анализ — исследование объектов предметной области.

2. Жизненный цикл программного продукта

Жизненный цикл - это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации. Он базируется на 3-х группах процессов:

1. **Основные процессы** — реализуются под управлением основных сторон (заказчик, поставщик, разработчик, оператор и персонал сопровождения), вовлеченных в жизненный цикл программных средств.
Примеры: заказ, поставка, разработка, эксплуатация, сопровождение.
2. **Вспомогательные процессы** — обеспечивают выполнение основных процессов.
Примеры: документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместный анализ, аудит, решение проблем.
3. **Организационные процессы** — применяются для создания, реализации и постоянного совершенствования основной структуры, охватывающей взаимосвязанные процессы жизненного цикла и персонал.
Примеры: управление, создание инфраструктуры, усовершенствование, обучение.

Стадии жизненного цикла: замысел → разработка → производство → применение → поддержка → списание.

3. Каскадная модель

Этапы каскадной модели:

1. определение требований;
2. проектирование;
3. кодирование, тестирование модулей;
4. интеграция, тестирования;
5. эксплуатация, сопровождение.

Характеристика модели:

- фиксированный набор стадий;
- каждая стадия — законченный результат;
- стадия начинается, когда закончилась предыдущая.

Недостатком модели является ее «негибкость»:

- фаза должна быть закончена, прежде чем приступить к следующей;
- набор фаз фиксирован;
- тяжело реагировать на изменение требований.

Каскадную модель **рекомендуется использовать** там, где требования заранее известны и неизменны.

4. Спиральная модель

Спиральная модель — частный случай итерационного подхода:

- вместо действий с обратной связью — спираль;
- отсутствуют заранее фиксированные фазы;
- каждый виток спирали — 1 итерация;
- каждый виток разбит на 4 сектора:
 - определение целей;
 - оценка и разрешение рисков;
 - разработка и тестирование;
 - планирование.
- на каждом витке могут применяться разные модели процесса разработки ПО.

Главное отличие — акцент на анализ и преодоление рисков.

5. Agile

Agile — это семейство «гибких» подход к разработке ПО. Agile предполагает, что при реализации проекта

- не нужно опираться только на заранее созданные подробные планы;
- важно ориентироваться на постоянно меняющиеся условия внешней и внутренней среды;
- учитывать обратную связь от заказчиков и пользователей.

Частными случаями agile-подходов являются *scrum* и *kanban*.

6. Kanban

Особенности:

- *kanban* — это «подход баланса»;
- задача — сбалансировать разных специалистов внутри команды и избежать ситуации, когда дизайнеры работают сутками, а разработчики жалуются на отсутствие новых задач;
- вся команда едина — отсутствуют роли владельца продукта и *scrum*-мастера;
- бизнес-процесс делится не на универсальные спринты, а на стадии выполнения конкретных задач (планируется, разрабатывается, тестируется, завершено);
- для визуализации используются доски (физ. и эл.) — они позволяют сделать рабочий процесс открытым и понятным для всех специалистов.

Практики:

- визуализация потока задач;
- ограничение невыполненных работ;
- управление рабочим потоком;
- использование явных правил;
- введение петли обратной связи;
- улучшение и эволюция.

Kanban отлично подходит для небольших проектов, бизнес вебсайтов, где не требуется много времени на планирование. Также он хорошо подходит для долгосрочных проектов, где нет четкой спецификации где задания формируются в процессе разработки.

7. Scrum

Спринт — итерация (цикл выпуска продукта):

- имеет фиксированную длительность, обычно от 2 до 8 недель;
- результат — готовый продукт, который потенциально можно передать заказчику;
- в течении спринта делаются все работы по сбору требований, дизайну, кодированию;
- рамки спринта фиксированы;
- каждый спринт начинается с собрания по его планированию и заканчивается собранием, где подводятся итоги спринта.

Scrum подходит для крупного проекта (длительность от 3 месяцев), который имеет полную спецификацию и требования перед началом разработки. В таком случае команда может составить детальный план разработки и весь процесс поделить на спринты.