



**Частное учреждение профессионального образования  
«Высшая школа предпринимательства (колледж)»  
(ЧУПО «ВШП»)**

**Кафедра информационных технологий**

**Внедрение и поддержка  
компьютерных систем**

# О чём этот междисциплинарный курс?

МДК 04.01 "Внедрение и поддержка компьютерных систем" формирует у будущих специалистов теоретические знания и практические навыки разработки программного обеспечения.

# Фундаментальные понятия

Система - сущность, которая в результате взаимодействия ее частей (компонентов) может поддерживать свое существование и функционировать как единое целое.

Системы функционируют как целое, что порождает у них свойства, отличающиеся от свойств составляющих ее частей.

# Фундаментальные понятия

Принято выделять простые и сложные системы.

Сложность может проявляться двумя различными путями: статическим и динамическим.

Статическая сложность (детальная) определяется в процессе детализации как количество рассматриваемых элементов.

Динамическая сложность зависит от отношений между элементами.

*Информационные системы относятся к классу динамически сложных.*

*Информационная система состоит из множества элементов или подсистем, которые находятся в разных состояниях и могут изменяться, в зависимости от изменения других частей.*

# Фундаментальные понятия

Любая техническая система создается под заранее известную цель, которая обычно является субъективной, поскольку она предлагается разработчиком, но эта цель должна исходить из объективных потребностей общества.

Таким образом, можно считать, что цель формируется в процессе взаимодействия между явлениями окружающей действительности. При этом возникает ситуация, которая заставляет строить новую систему.

# Фундаментальные понятия

Жизненный цикл информационной системы – период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент ее полного изъятия из эксплуатации.



# Жизненный цикл информационной системы

**Жизненный цикл разработки ИС (ПО)** (*System/Software Development Life Cycle, SDLC*) — процесс, состоящий из конкретных этапов, который начинается в момент принятия решения о необходимости создания ИС (П)О и заканчивается в момент прекращения поддержки ИС(ПО) разработчиками

# **Зачем нам знать «Жизненный цикл информационной системы»**

Визуализировать сложный процесс разработки

Управлять проектом

Предсказывать и планировать доставку рабочих продуктов в ходе всего процесса разработки

Управлять рисками выхода за рамки бюджета / превышения срока реализации

Быстро определять, на каком этапе находится разработка в данный момент



# Этапы жизненного цикла разработки

Всего выделяют 7 основных этапов разработки:

Идея

Определение требований

Дизайн (архитектура) системы

Разработка

Тестирование

Развертывание

Поддержка

Этап закрытия не является обязательным и зависит от проекта.

# Этапы жизненного цикла разработки



# Подробнее про этап «Тестирование»

**Тестирование ПО (Software Testing) — проверка соответствия между реальным и ожидаемым поведением программы, проводится на наборе тестов, который выбирается некоторым образом.**

Чем занимаются в тестировании:

1. планированием работ (Test Management)
2. проектированием тестов (Test Design) — этап, на котором создаются тестовые сценарии (тест кейсы), в соответствии с определёнными ранее критериями. Т.е., определяется, КАК будет тестироваться продукт.
3. выполнением тестирования (Test Execution)
4. анализом результатов (Test Analysis)

# Для чего проводится тестирование ИС(ПО)?

Для проверки соответствия требованиям.

Для обнаружение проблем на более ранних этапах разработки и предотвращение

Повышения стоимости продукта.

Обнаружение вариантов использования, которые не были предусмотрены при разработке.

А также взгляд на продукт со стороны пользователя.

(Повышение лояльности к компании и продукту, т.к. любой обнаруженный дефект негативно влияет на доверие пользователей.)

# Принципы тестирования

**Принцип 1 — Тестирование демонстрирует наличие дефектов (Testing shows presence of defects).** *Тестирование только снижает вероятность наличия дефектов, которые находятся в программном обеспечении, но не гарантирует их отсутствия.*

**Принцип 2 — Исчерпывающее тестирование невозможно (Exhaustive testing is impossible).** *Полное тестирование с использованием всех входных комбинаций данных, результатов и предусловий физически невыполнимо (исключение — тривиальные случаи).*

# Принципы тестирования(1)

**Принцип 3 — Раннее тестирование (Early testing).** *Следует начинать тестирование на ранних стадиях жизненного цикла разработки ПО, чтобы найти дефекты как можно раньше.*

**Принцип 4 — Скопление дефектов (Defects clustering).** *Большая часть дефектов находится в ограниченном количестве модулей.*

**Принцип 5 — Парадокс пестицида (Pesticide paradox).** *Если повторять те же тестовые сценарии снова и снова, в какой-то момент этот набор тестов перестанет выявлять новые дефекты.*

# Принципы тестирования(2)

**Принцип 6 — Тестирование зависит от контекста (Testing is context depending).** *Тестирование проводится по-разному в зависимости от контекста. Например, программное обеспечение, в котором критически важна безопасность, тестируется иначе, чем новостной портал.*

**Принцип 7 — Заблуждение об отсутствии ошибок (Absence-of-errors fallacy).** *Отсутствие найденных дефектов при тестировании не всегда означает готовность продукта к релизу. Система должна быть удобна пользователю в использовании и удовлетворять его ожиданиям и потребностям.*