

# Современная методология и разработка по

СДЕЛАЛИ: ГАДЖИЕВ АСИМ,  
АХТАРЬЯНОВ ДИНИСЛАМ

20п-3



# Этапы жизненного цикла ПО

У любого программного обеспечения есть жизненный цикл — этапы, через которые оно проходит с начала создания до конца разработки и внедрения. Чаще всего это подготовка, проектирование, создание и поддержка.

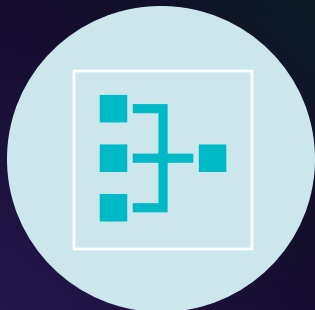
# Рассмотрим эти этапы на примере жизненного цикла на примере автосалона.



**Подготовка.** Пётр решил запустить интернет-автосалон и начал анализировать, какие подобные сайты уже представлены в сети. Собрал информацию об их трафике, функциональности.



**Проектирование.** Пётр выбрал компанию-подрядчика и обсудил с её специалистами архитектуру и дизайн будущего интернет-автосалона.



**Создание.** Пётр заключил с разработчиками договор. Они начали писать код, отрисовывать дизайн, составлять документацию.



**Поддержка.** Пётр подписал акт сдачи-приёмки, и подрядчик разместил интернет-автосалон на серверах. Пользователи начали его посещать и сообщать о замеченных ошибках в поддержку, а программисты — оперативно всё исправлять.

# Основные модели разработки ПО

- Code and fix – модель кодирования и устранения ошибок;
- Waterfall Model – каскадная модель, или «водопад»;
- V-model – V-образная модель, разработка через тестирование;
- Incremental Model – инкрементная модель;
- Iterative Model – итеративная (или итерационная) модель;
- Spiral Model – спиральная модель;
- Chaos model – модель хаоса;
- Prototype Model – прототипная модель



# Waterfall (каскадная модель, или «водопад»)

В этой модели разработка осуществляется поэтапно: каждая следующая стадия начинается только после того, как заканчивается предыдущая.



# V-образная модель (разработка через тестирование)

Это усовершенствованная каскадная модель, в которой заказчик с командой программистов одновременно составляют требования к системе и описывают, как будут тестировать её на каждом этапе.




# Incremental Model (инкрементная модель)

В инкрементной модели разработка по частям. Целью программиста является разработка основных функций. А затем протестировать на пользователях.

Если заказчику и пользователям нравится продукт, то команда разработки продолжают работу, но уже по частям, тем самым совершенствуя продукт, приближаясь к описанному в техническом задании







# Iterative Model (итеративная модель)

Эта модель похожа на инкрементную но в отличии от нее итеративная не имеет технического задания. Команда разработчиков вместе с заказчиком выставляют продукт в магазин и уже в зависимости от требований пользователей дорабатывают





# Spiral Model (спиральная модель)

Используя эту модель, заказчик и команда разработчиков серьёзно анализируют риски проекта и выполняют его итерациями. Последующая стадия основывается на предыдущей, а в конце каждого витка — цикла итераций — принимается решение, продолжать ли проект. Эту модель начали использовать в 1988 году.

# Agile - подход к разработке

Agile («эджайл») переводится с английского как «гибкий». Включает в себя практики, подходы и методологии, которые помогают создавать продукт более эффективно, а именно:





```

mirror_mod = modifier_ob.
set mirror object to mirror
mirror_mod.mirror_object

operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob
mirror_ob.select = 0
= bpy.context.selected_object
data.objects[one.name].select
print("please select exactly

-- OPERATOR CLASSES -----

types.Operator):
on X mirror to the selected
object.mirror_mirror_x"
mirror X"

context):
context.active_object is not

```

- экстремальное программирование (Extreme Programming, XP);
- бережливую разработку программного обеспечения (Lean);
- фреймворк для управления проектами Scrum;
- разработку, управляемую функциональностью (Feature-driven development, FDD);
- разработку через тестирование (Test-driven development, TDD);
- методологию «чистой комнаты» (Cleanroom Software Engineering);
- итеративно-инкрементальный метод разработки (OpenUP);
- методологию разработки Microsoft Solutions Framework (MSF);
- метод разработки динамических систем (Dynamic Systems Development Method, DSDM);
- метод управления разработкой Kanban.



# Kanban

Сегодня это одна из наиболее популярных методологий разработки ПО. Команда ведёт работу с помощью виртуальной доски, которая разбита на этапы проекта. Каждый участник видит, какие задачи находятся в работе, какие — застряли на одном из этапов, а какие уже дошли до его столбца и требуют внимания.

В отличие от скрама, в канбане можно взять срочные задачи в разработку сразу, не дожидаясь начала следующего спринта.