

Этапы разработки и внедрения информационной системы.

Ситников Артём
ГФ25-02Б

Жизненный цикл информационной системы



ИС — это не просто ПО, а система людей, процессов и технологий.

Её создание требует чёткого соблюдения этапов жизненного цикла.

Этапы 1-3: От идеи до архитектуры

1) Планирование

- Бизнес-кейс, рамки, риски, утверждение проекта.



2) Анализ требований

- Функциональные и нефункциональные требования.

- Резуль

3) Проектирование

- Архитектура, БД, интерфейсы, API.
- Результат: Техническое

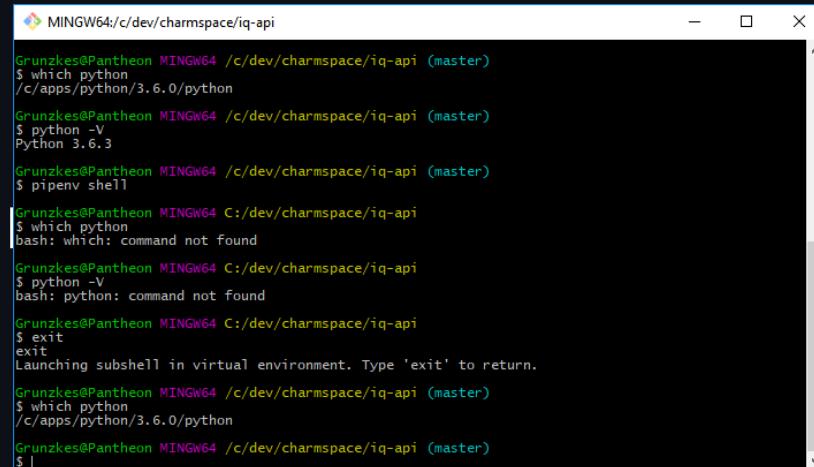
умент.



Этапы 4-5: Разработка и тестирование

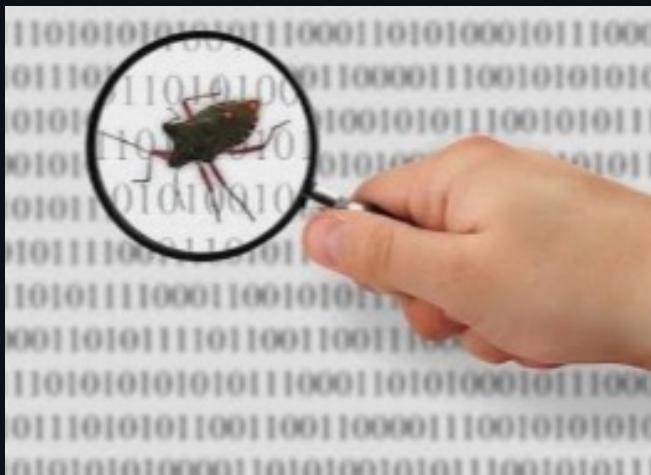
Реализация:

- Код, инфраструктура, интеграции
- Используются Git, CI, code review



```
Grunzkes@Pantheon MINGW64 /c/dev/charmspace/iq-api (master)
$ which python
/c/apps/python/3.6.0/python
Grunzkes@Pantheon MINGW64 /c/dev/charmspace/iq-api (master)
$ python -V
Python 3.6.3
Grunzkes@Pantheon MINGW64 /c/dev/charmspace/iq-api (master)
$ pipenv shell
Grunzkes@Pantheon MINGW64 C:/dev/charmspace/iq-api
$ which python
bash: which: command not found
Grunzkes@Pantheon MINGW64 C:/dev/charmspace/iq-api
$ python -V
bash: python: command not found
Grunzkes@Pantheon MINGW64 C:/dev/charmspace/iq-api
$ exit
exit
Launching subshell in virtual environment. Type 'exit' to return.
Grunzkes@Pantheon MINGW64 /c/dev/charmspace/iq-api (master)
$ which python
/c/apps/python/3.6.0/python
Grunzkes@Pantheon MINGW64 /c/dev/charmspace/iq-api (master)
$ |
```

Тестирование



Вид	Описание
Модульное (unit)	Проверка отдельных компонентов
Интеграционное	Проверка взаимодействия модулей
Системное	Проверка всей системы как единого целого
Приёмочное (UAT)	Тестирование конечными пользователями

Внедрение и сопровождение

Внедрение включает:

- Миграцию данных
- Обучение пользователей
- Пилотный или полный запуск



ЧТО НОВОГО

- пофиксили баги
- поели шашлыков

Вот даже не знаем, что важнее

Стратегии внедрения:

- Параллельная; Постепенная; «Big Bang»

```
#include "СлавянскийС++.h"
// пишем первую программу на втт
// пишем "славься Русь!"

внедрить хутор Русь;

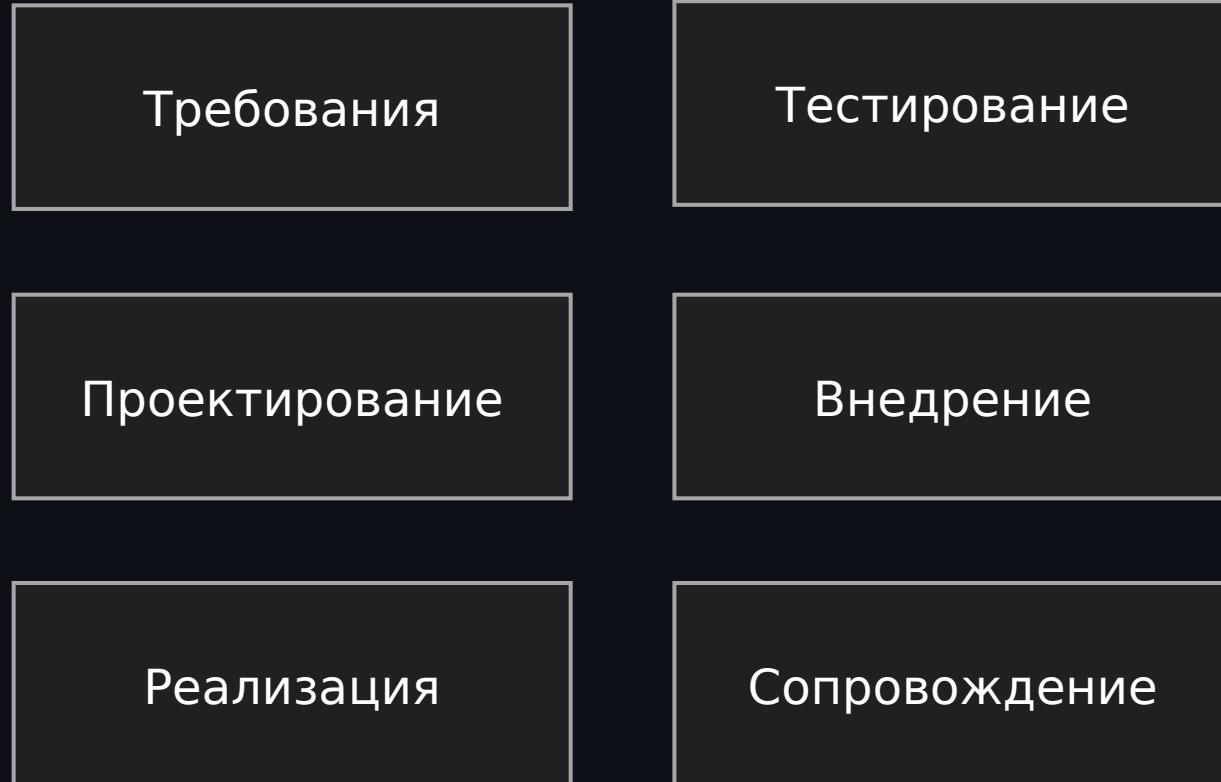
царь_батюшка_главный()
{
    вперёд_славяне;
    строка строкенка = "Славься Русь!";
    молвить << строкенка << прыг_скок;
    воздать 0;
}
```

Сопровождение:

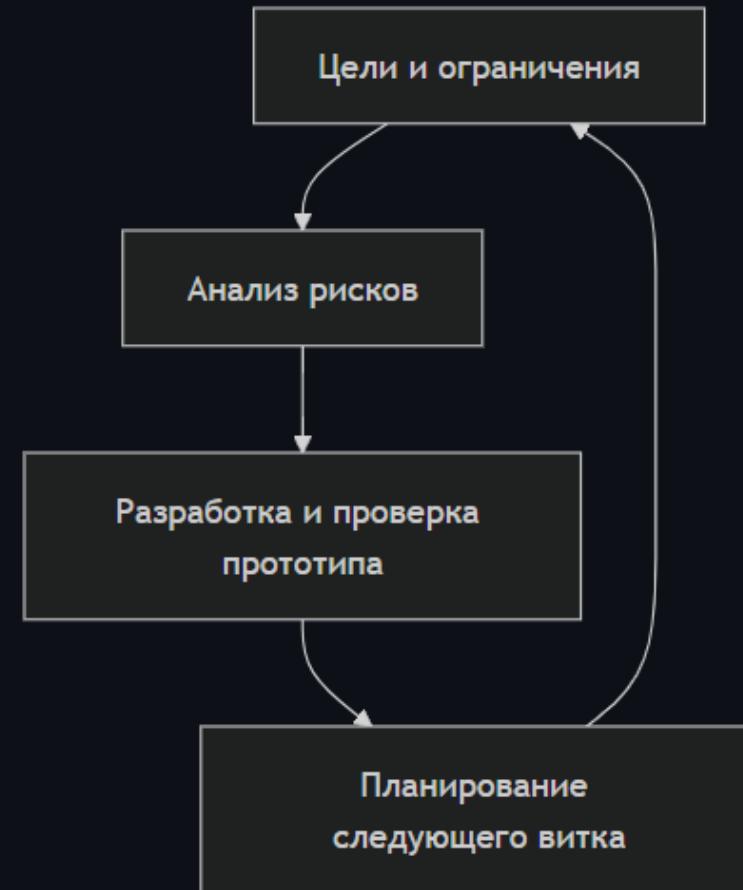
- Исправление ошибок
- Обновления и улучшения
- Обратная связь → новые версии

Модели жизненного цикла

Каскадная (Waterfall)



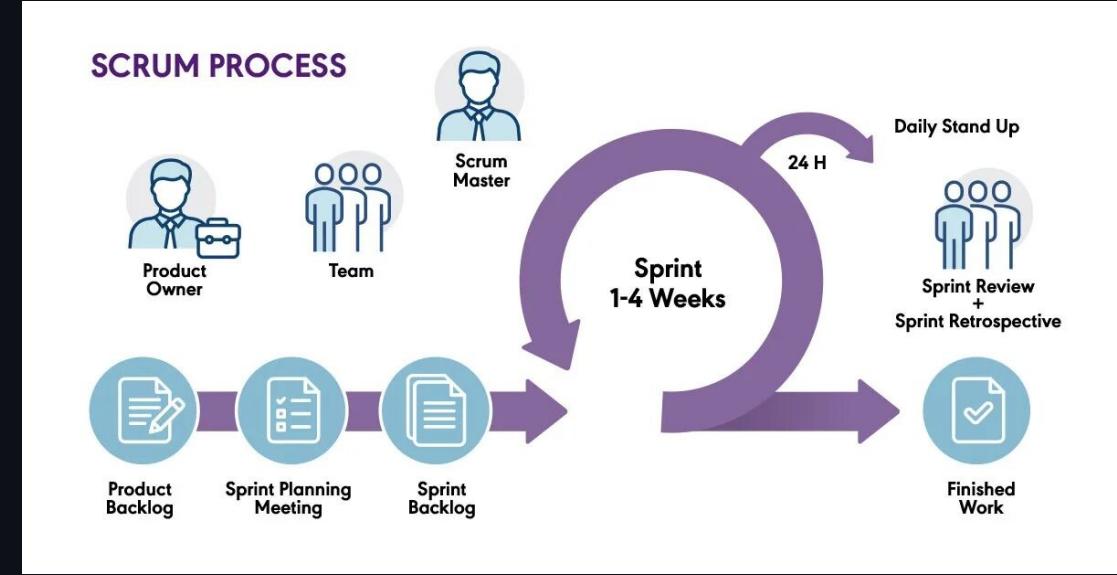
Спиральная



Гибкие методологии (Agile/Scrum)

Ключевые принципы:

- Работающее ПО важнее исчерпывающей документации
- Готовность к изменениям даже на поздних этапах
- Регулярные демонстрации результатов



Роли в Scrum:

Роль	Обязанности
Product Owner	Формирует бэклог, ставит приоритеты
Scrum Master	Обеспечивает процесс, устраняет блокировки
Команда	Разрабатывает и тестирует функционал

Факторы успеха ИС

Фактор	Почему важен
Чёткие цели и требования	Снижает риск переделок и срывов сроков
Участие пользователей	Обеспечивает соответствие системы реальным нуждам
Поддержка руководства	Гарантирует выделение ресурсов и приоритетность проекта
Квалифицированная команда	Качество зависит от экспертизы разработчиков и аналитиков
Управление изменениями	Помогает адаптировать организацию к новой системе
Обучение пользователей	Повышает принятие системы и снижает сопротивление

Технологии — лишь 30% успеха. Люди и процессы — 70%.

Заключение

Разработка ИС – это трансформация бизнеса.

Современные проекты часто используют гибридные подходы:

→ структура Waterfall + скорость Agile



Успех = глубокий анализ + грамотное проектирование + гибкое управление