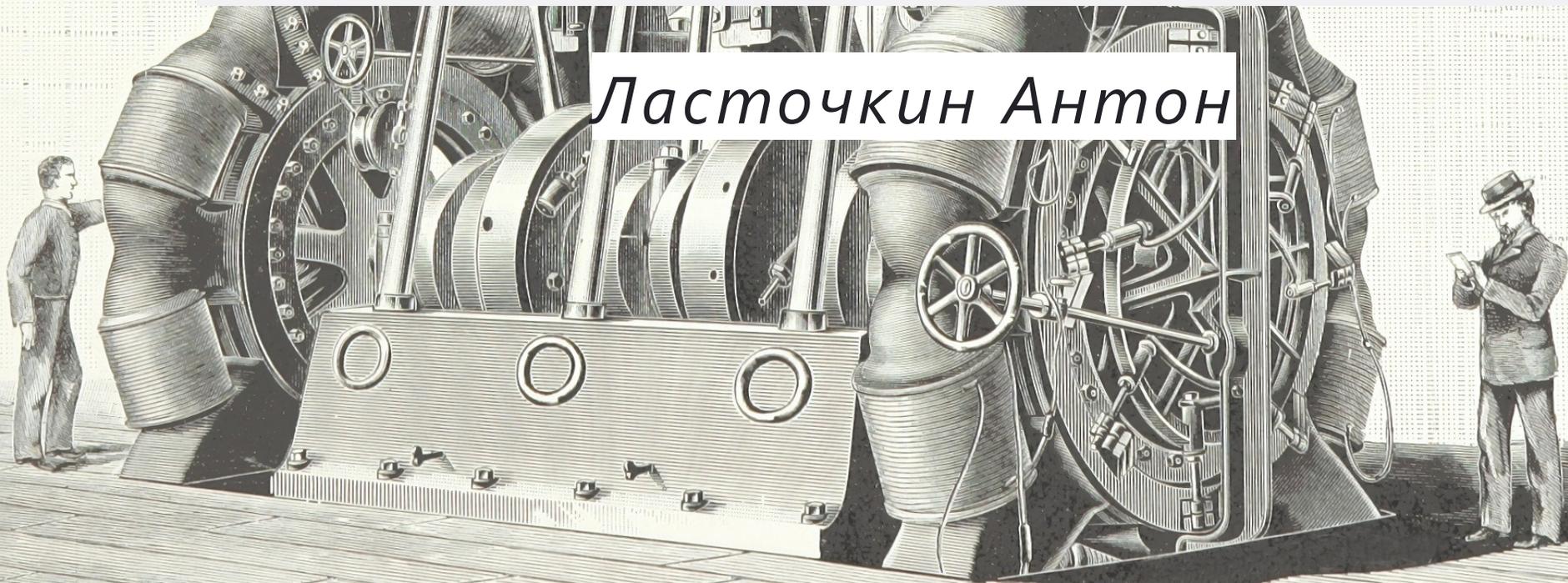


# Мониторим FSM во встраиваемом ПО

*Ласточкин Антон*





- Разработчик Embedded с опытом 15+
- Дополнительно: CV/Qt/backend
- Преподаватель на edtech-площадках  
Подробнее:  
<http://antlas.ru>

# План доклада

- КА и дроны

# План доклада

- КА и дроны
- Use-case на примере включения нагрузки

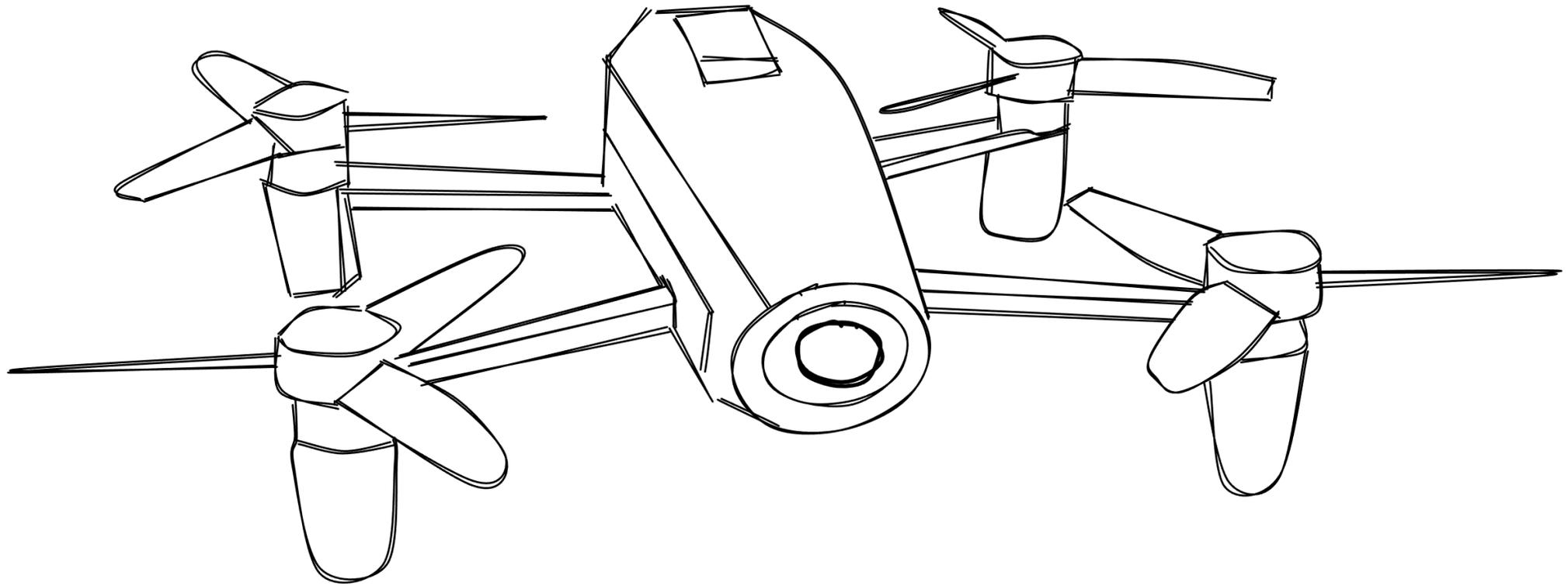
# План доклада

- КА и дроны
- Use-case на примере включения нагрузки
- Кодим без разметки, двигаем требования

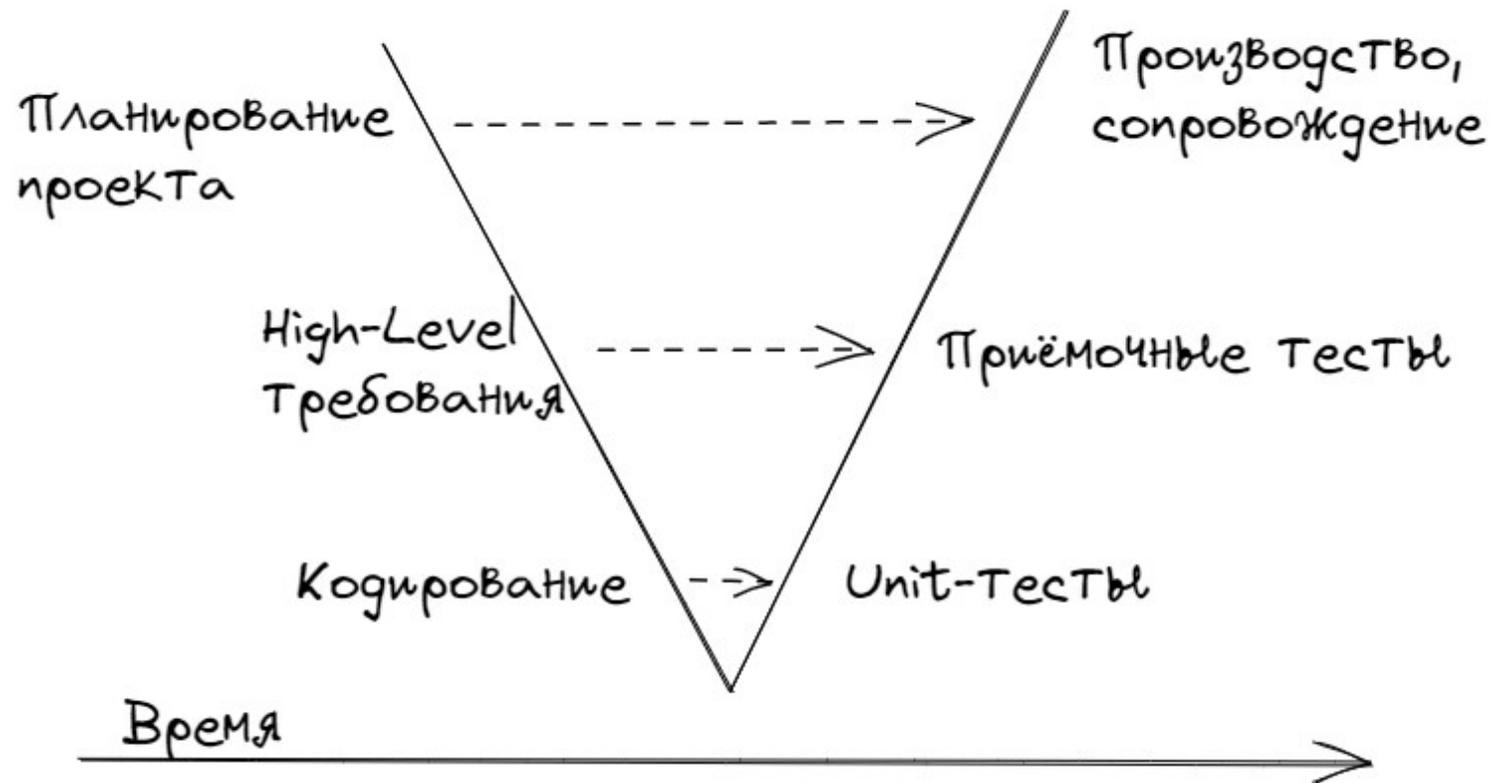
# План доклада

- КА и дроны
- Use-case на примере включения нагрузки
- Кодим без разметки, двигаем требования
- Тот же код с разметкой
- Генерируем графы, анализируем

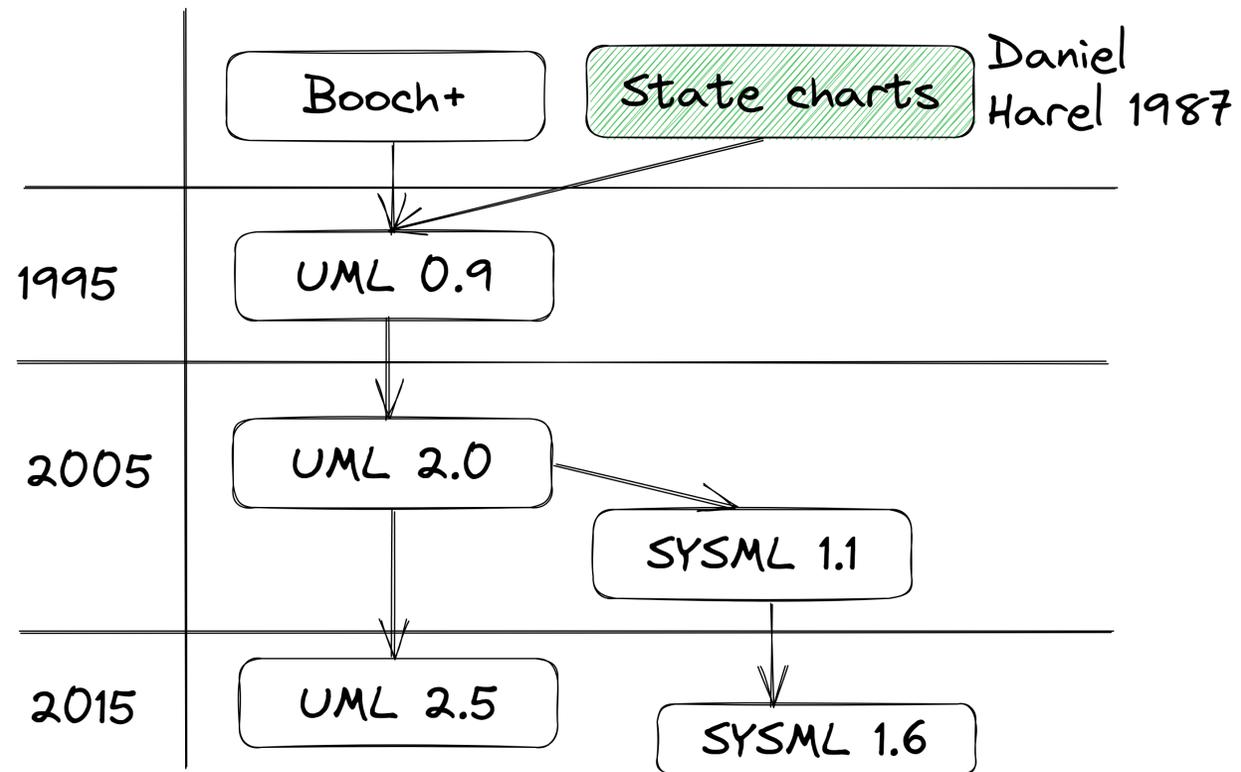
# Немного о дронах



# Модель разработки



# FSM и UML



# Инженерный подход

- SCADA state machines
- Matlab stateflow
- Labview

# Начальное ТЗ

$N_b$  - средняя частота вращения двигателей

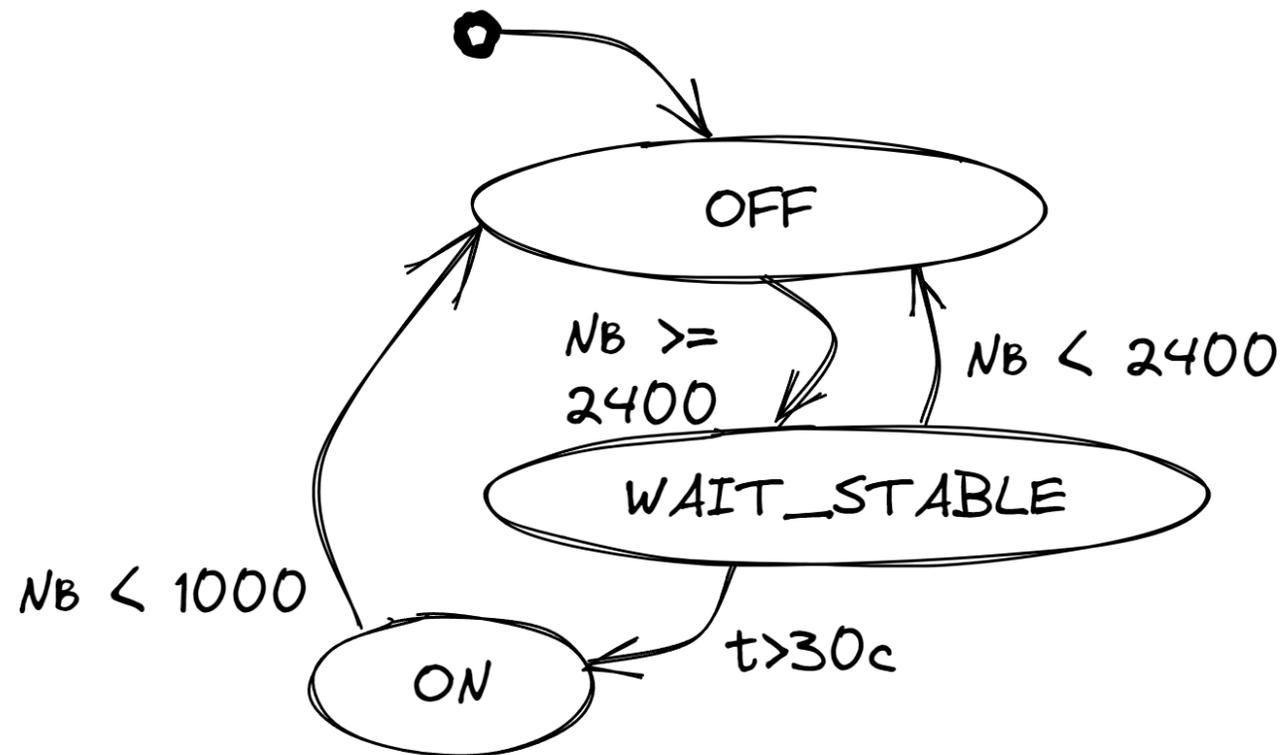
1. Включение нагрузки выполнять при  $N_b \geq 2400$  об/мин установившегося в течение 30 секунд.
2. Если течение 30 секунд после включения  $N_b < 2400$  об/мин, отсчёт времени должен прерываться.
3. Выключение нагрузки - при  $N_b < 1000$  об/мин.

# ТЗ, код и графы

<https://antlas.ru/fsm>



# FSM по требованиям



# Код switch-case

```
switch (state)
{
case PowerState::Off:
    if (freq >= ON_RPM) {
        state = PowerState::WaitStable;
        start_wait = curr_ms;
    }
    break;
case PowerState::WaitStable:
    if (freq < ON_RPM)
    {
        start_wait = 0;
        state = PowerState::Off;
    }
    else if ( (curr_ms - start_wait) >= MAX_WAIT_STABLE_MS)
    {
        start_wait = 0;
        state = PowerState::On;
    }
    break;
case PowerState::On:
    if (freq <= OFF_RPM)
    {
        state = PowerState::Off;
    }
    break;
}
```

# Усложнение ТЗ

- Новые признаки
- Раздельная логика для земли и воздуха
- Учёт стабильности параметров

# Три точки зрения



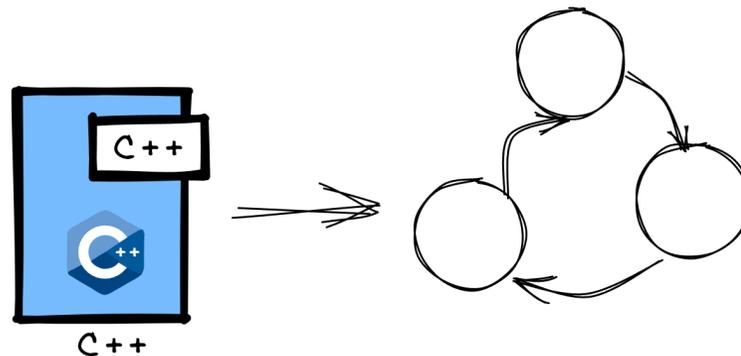
# Типичные варианты решения

1. Тщательное комментирование
2. Генерация кода автомата по спецификации
3. Новый способ представления в коде

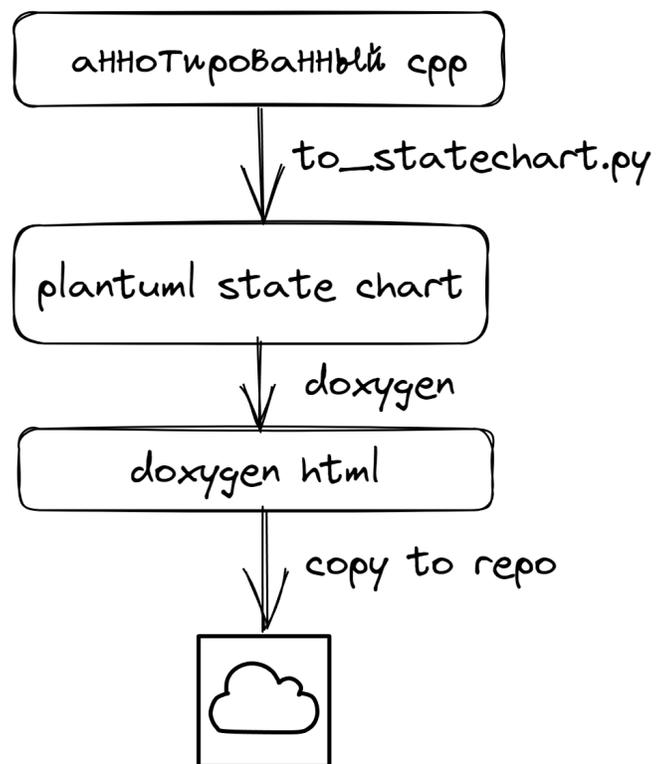
# Строим документацию

## Тулинг:

- Питон скрипт
- doxygen
- plantuml



# Пайплайн



# Код с аннотацией

```
//@statechart dia_turn_on Диаграмма включения полезной нагрузки
//@Условия:
//@ - ОБ - Частота оборотов от двух датчиков
//@ - t - Время со старта блока
//@[*]-->Off

switch (state)
{
case PowerState::Off: //@Off: Нагрузка выключена
    if (freq >= ON_RPM) { //@Off-->WaitStable: ОБ>=2400
        ...
    }
}
```

# Plantuml

```
@startuml
[*]-->Off
Off: Нагрузка выключена
Off-->WaitStable: ОБ>=2400
WaitStable: Ожидание стабильности
WaitStable-->Off: ОБ<2400
WaitStable-->On: t >= 30с
On: Нагрузка включена
On-->Off: ОБ<1000
```

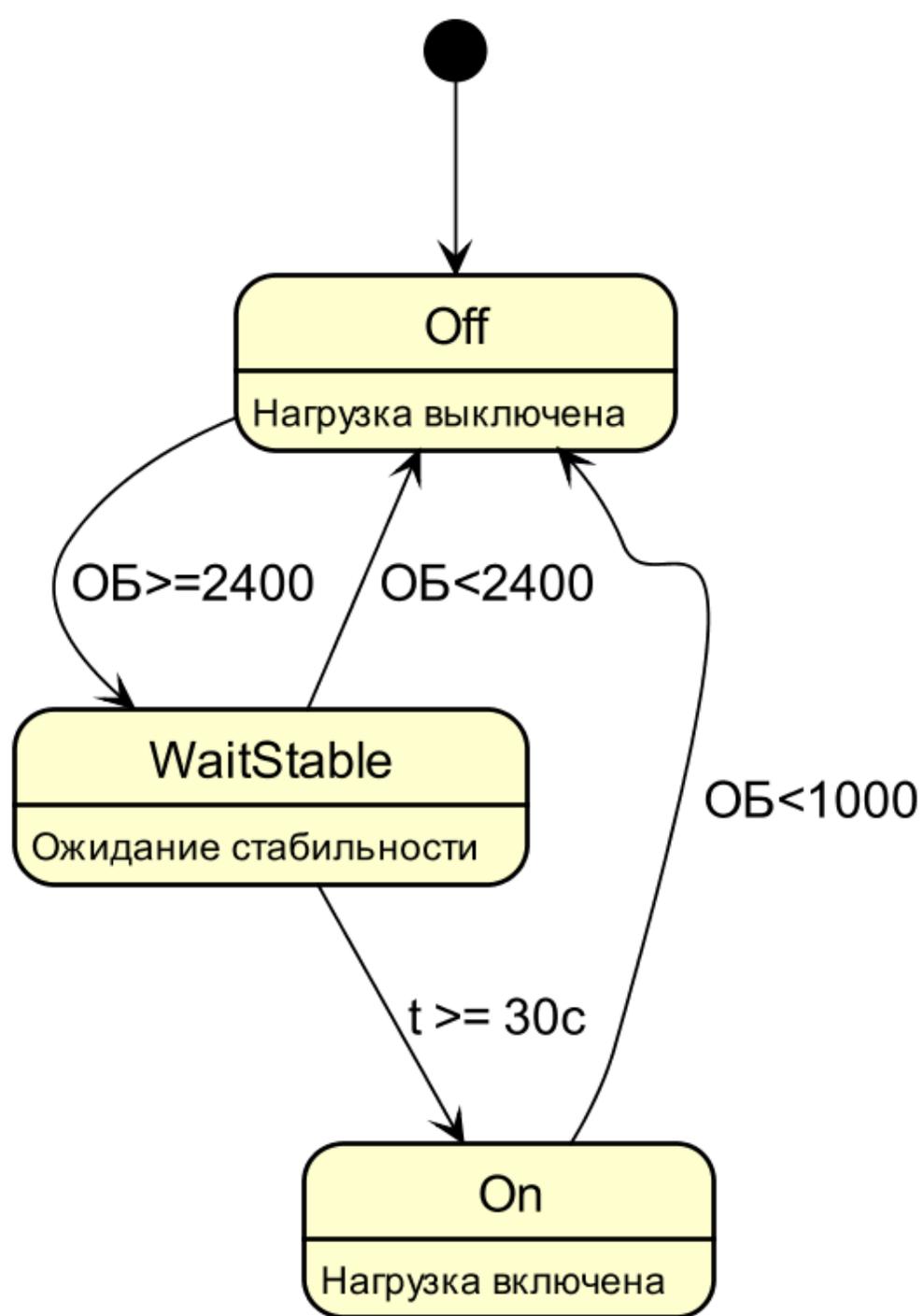
legend

Условия:

- ОБ - Усреднённая частота оборотов
- t - Время со старта блока

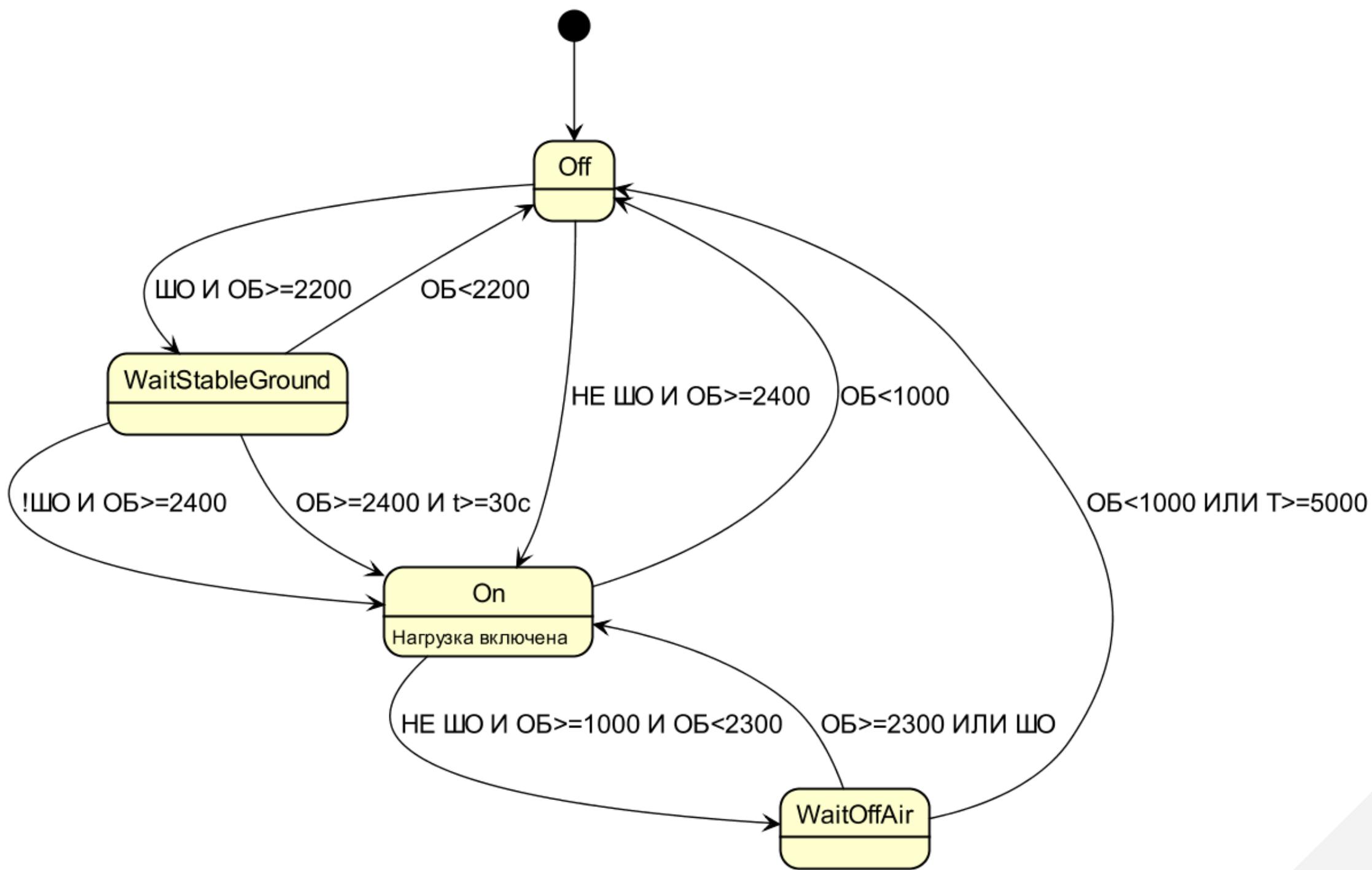
endlegend

```
@enduml
```



# Новые требования

1. Включение нагрузки выполнять при  $N_v \geq 2200$  об/мин установившегося в течение 30 секунд для ШО.
3. Если течение 30 секунд после включения на земле  $N_v < 2200$  об/мин, отсчёт времени должен прерываться.
4. Выключение нагрузки - при  $N_v < 1000$  об/мин.
5. Включение нагрузки в воздухе выполнять без задержки при значении оборотов  $N_v \geq 2400$  об/мин
7. Если в течение 5 секунд при работе в воздухе обороты держаться ниже 2300, выключать нагрузку. При повышении оборотов  $N_v \geq 2400$  об/мин, сбрасывать таймер.



# Ньюансы в тестировании

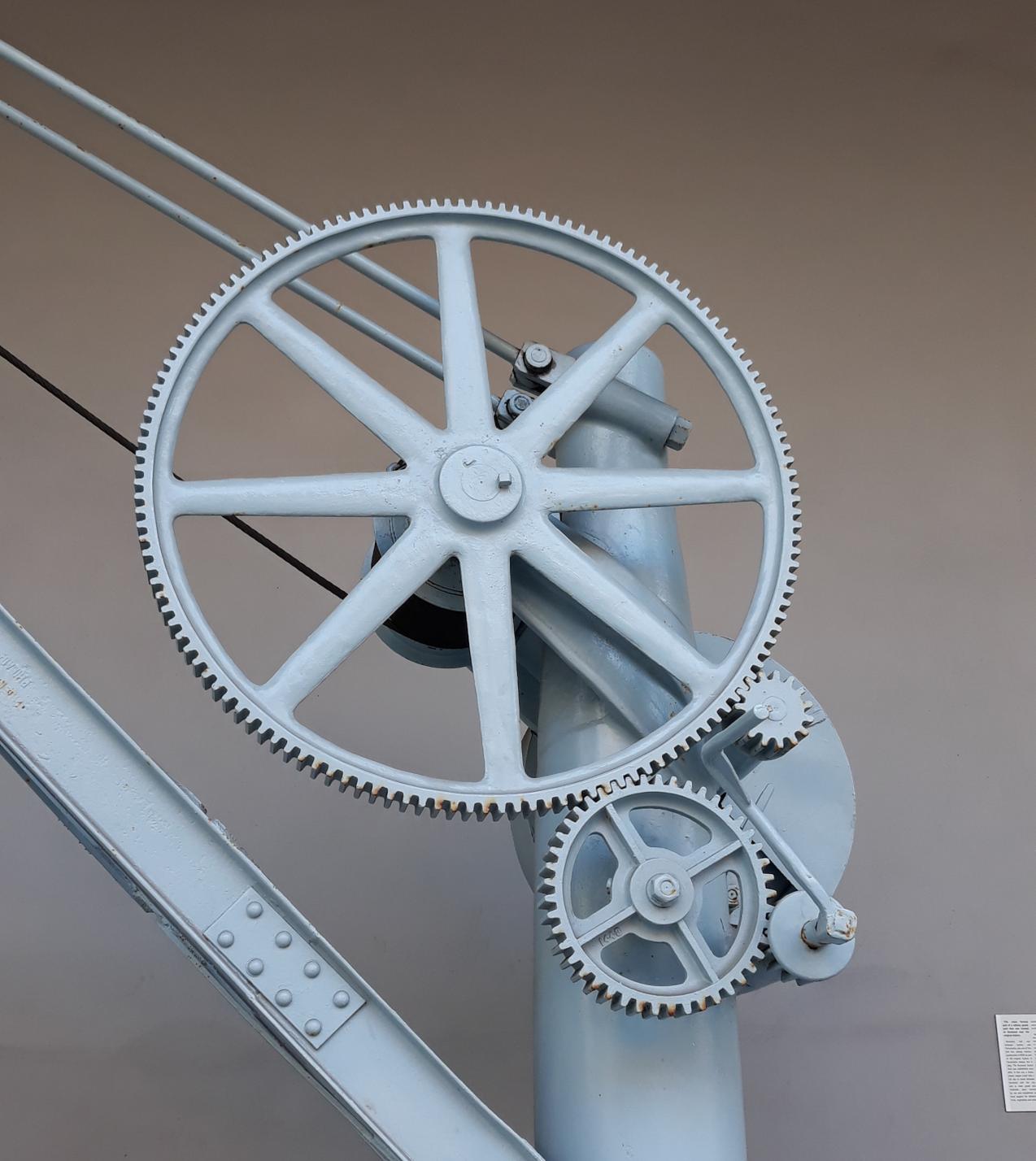
- Добавляем печать переходов с именами состояний из сгенерированного кода.

Пример лога:

```
100: N=0, WOW, power=0
200: N=2200, WOW, power=0
(::Power off -> wait on gnd)
30199: N=2200, WOW, power=0
30200: N=2200, WOW, power=1
(::Power wait on gnd -> on)
```

# Недостатки подхода

1. Непрерывная поддержка комментариев
2. Специальное форматирование
3. Выбор уровня абстракции для признаков
4. Дополнительный инструментарий



# Путь к успеху

- Выделение машин состояний в коде
- Актуальный граф переходов
- В общении находим верную модель