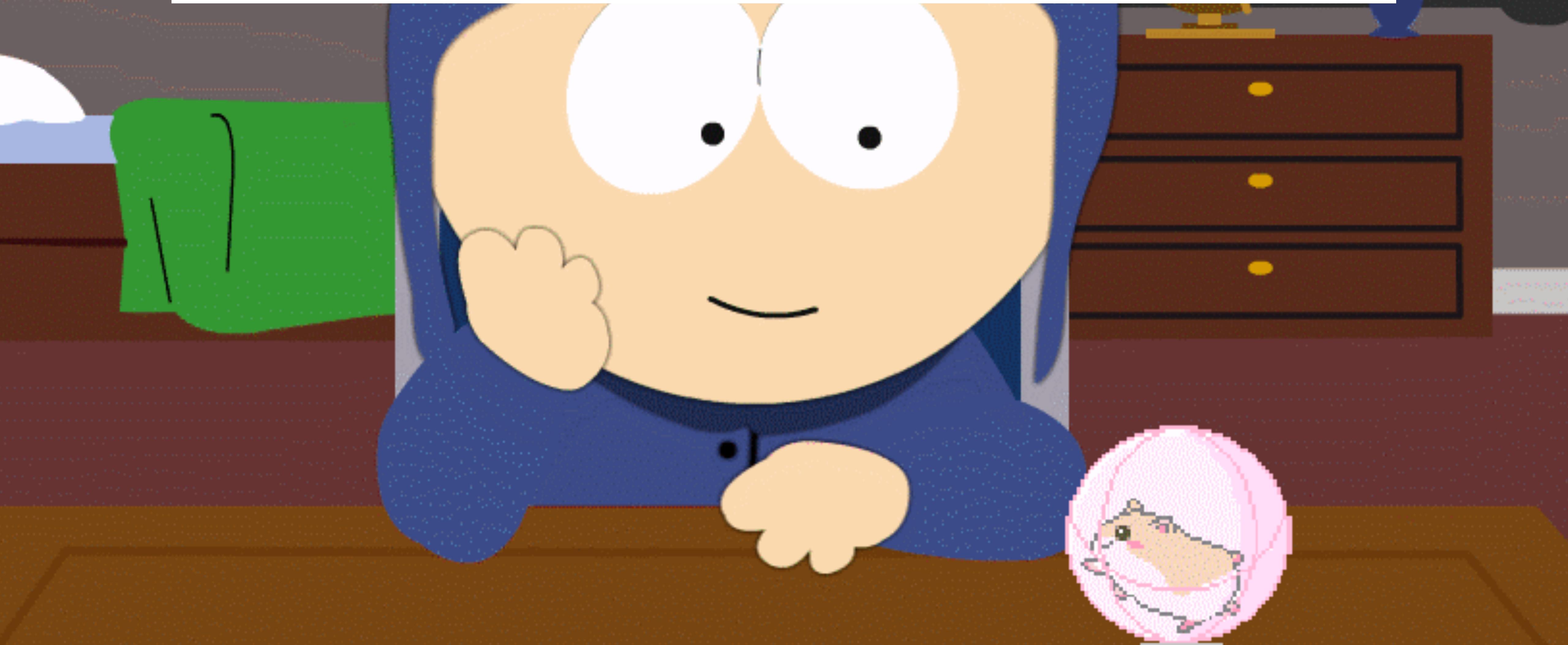


**Не повторяй моих ошибок
при подготовке доклада**



Всеволод БРЕКЕЛОВ

Разработчик



ПК @Heisenbug, @JPoint, @Joker

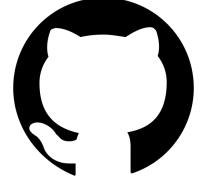
volekerb



brekelov



volekerb

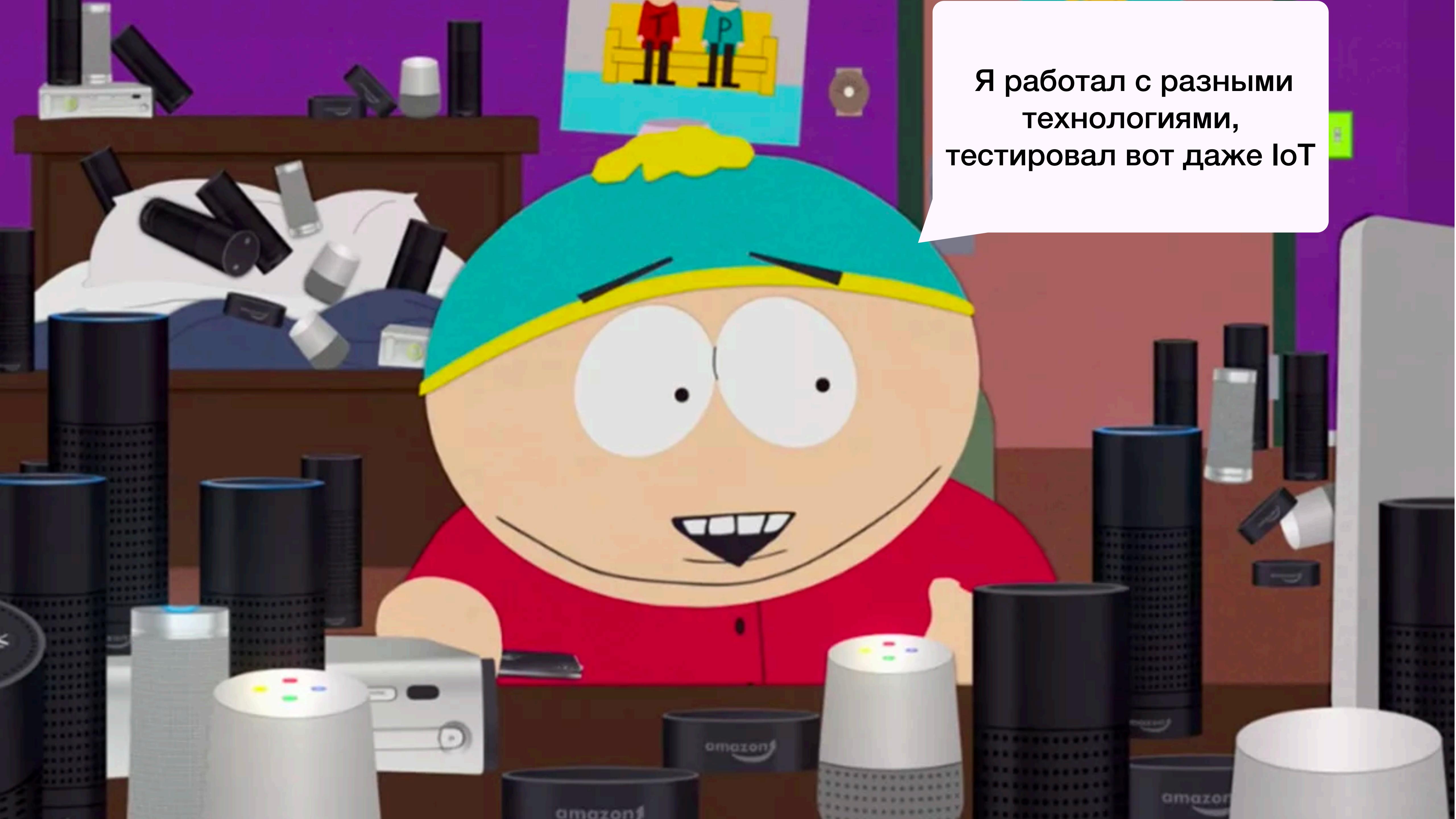


Обычная история

Любые совпадения случайны

И вообще это все про меня





Я работал с разными
технологиями,
тестировал вот даже IoT



Eric Cartman, a character from the animated TV show South Park, is shown in a room filled with numerous Amazon Echo smart speakers. He has a wide-eyed, slightly confused expression. A white speech bubble originates from his mouth, containing the text "Есть о чем рассказать..." (There's something to tell...). The room has purple walls, a wooden shelf with various items, and a painting of two figures on the wall.

Есть о чем рассказать...

A cartoon illustration of Eric Cartman from South Park. He is wearing a red shirt and a blue baseball cap with yellow stripes. He has a wide-eyed, slightly worried expression. He is surrounded by numerous Amazon Echo devices of various models and colors (black, white, grey) arranged in a grid pattern. In the background, there's a shelf with more tech gadgets like a laptop and a printer, and a small screen showing two people standing on a wooden platform.

Да и менеджер говорит,
что надо идти
выступать...

A cartoon illustration of Eric Cartman from South Park. He is wearing his signature green baseball cap and red shirt, with a wide-eyed, slightly smug expression. He is surrounded by numerous Amazon Echo smart speakers of various models and colors (black, white, grey) scattered throughout the scene. In the background, there's a shelf with more tech gadgets like a laptop and a printer, and a small screen showing two figures. A speech bubble originates from Eric's mouth.

И коллеги..



Eric Cartman from South Park is shown in a room filled with Amazon Echo smart speakers. He has a wide-eyed, excited expression. A speech bubble above him contains the text "Надо попробовать!" (I need to try it!).

Надо попробовать!

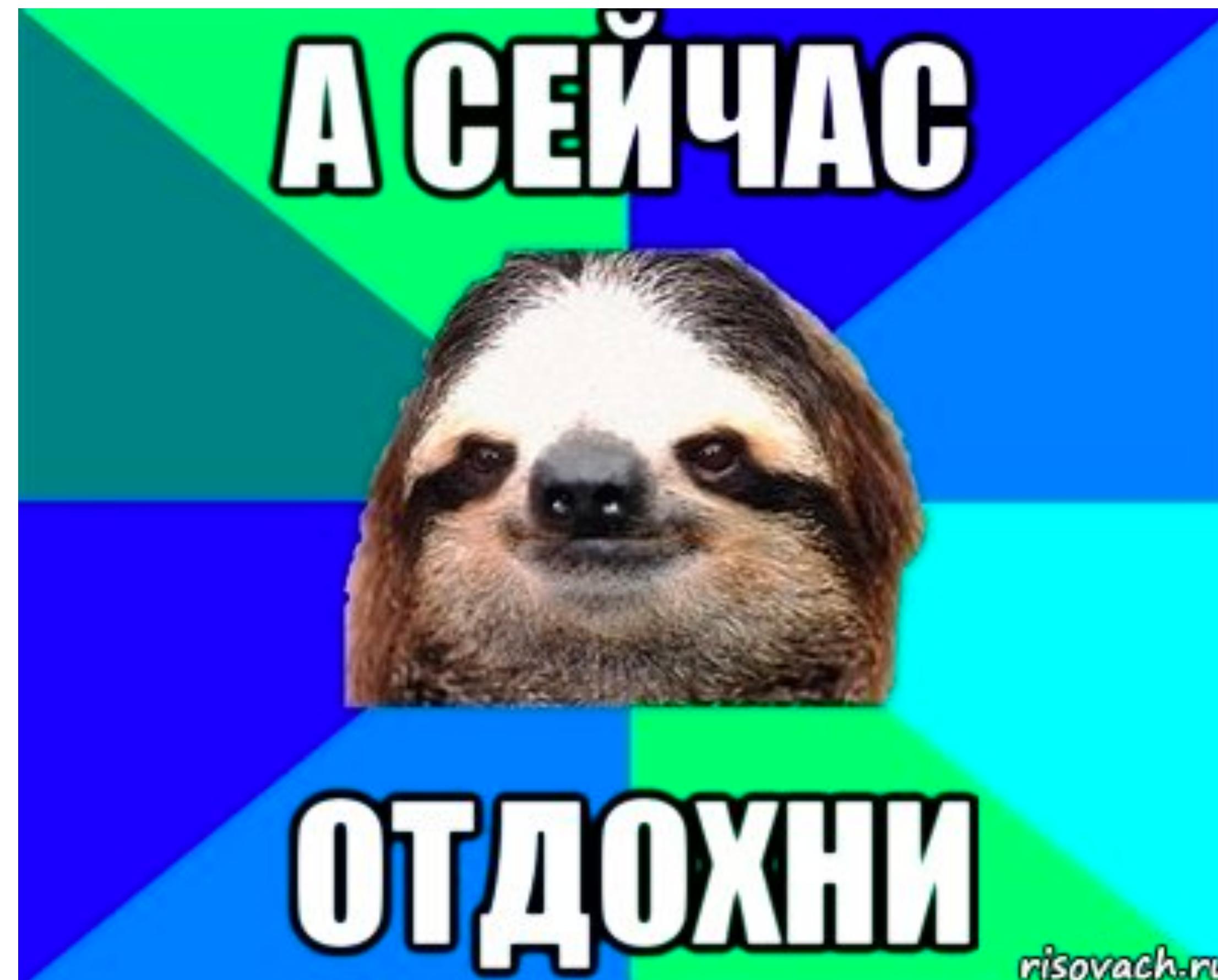
Февраль

Call For papers Open!

Конференция в июне

Времени так-то много еще

Сейчас так много дел...



ДВЕ НЕДЕЛЬКИ ОТДОХНУЛ

ЕЩЁ ДВЕ ОТДОХНИ

Середина апреля

Конференция через полтора месяца





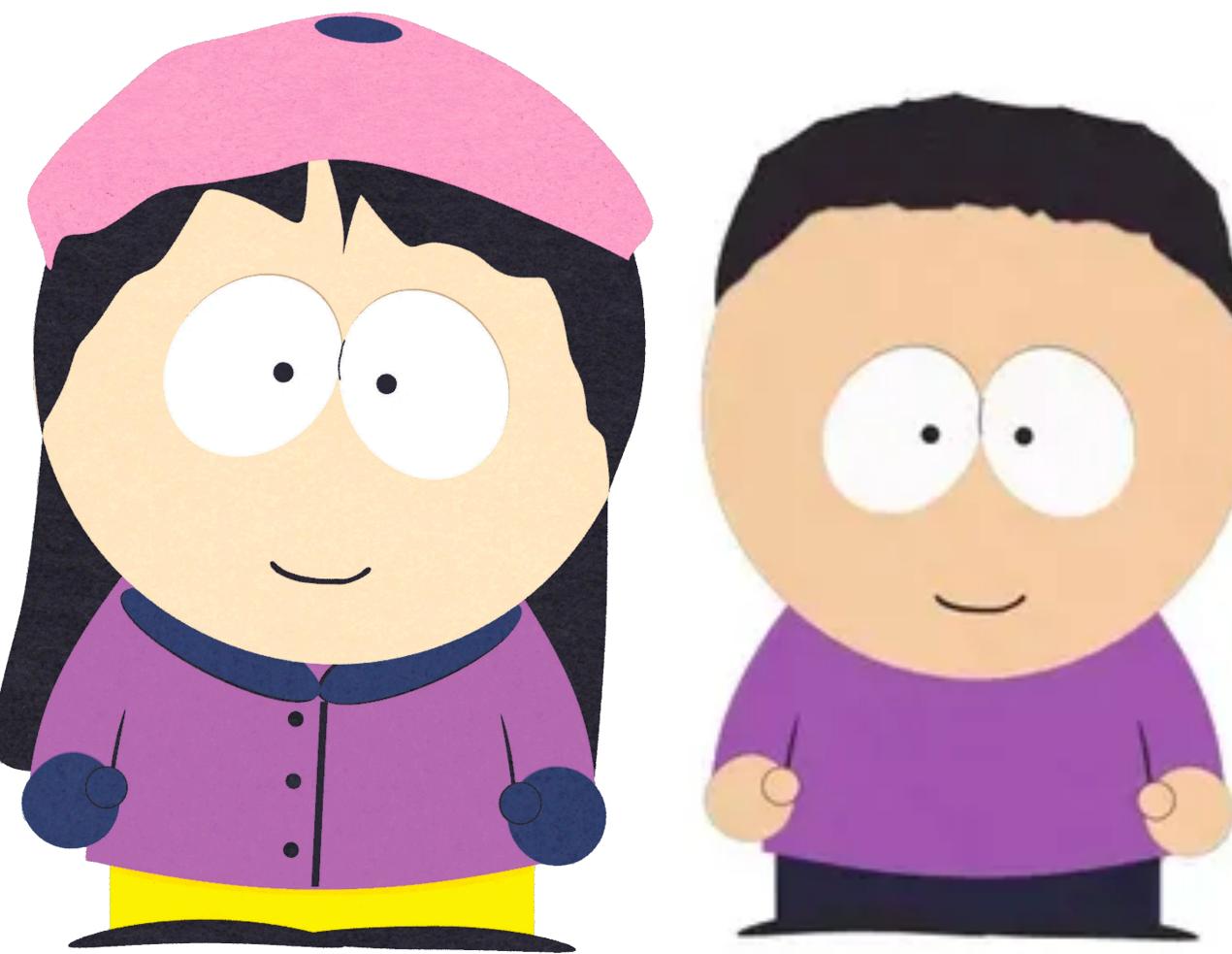
80% докладов уже принято

**А еще даже не понял,
что именно рассказать**



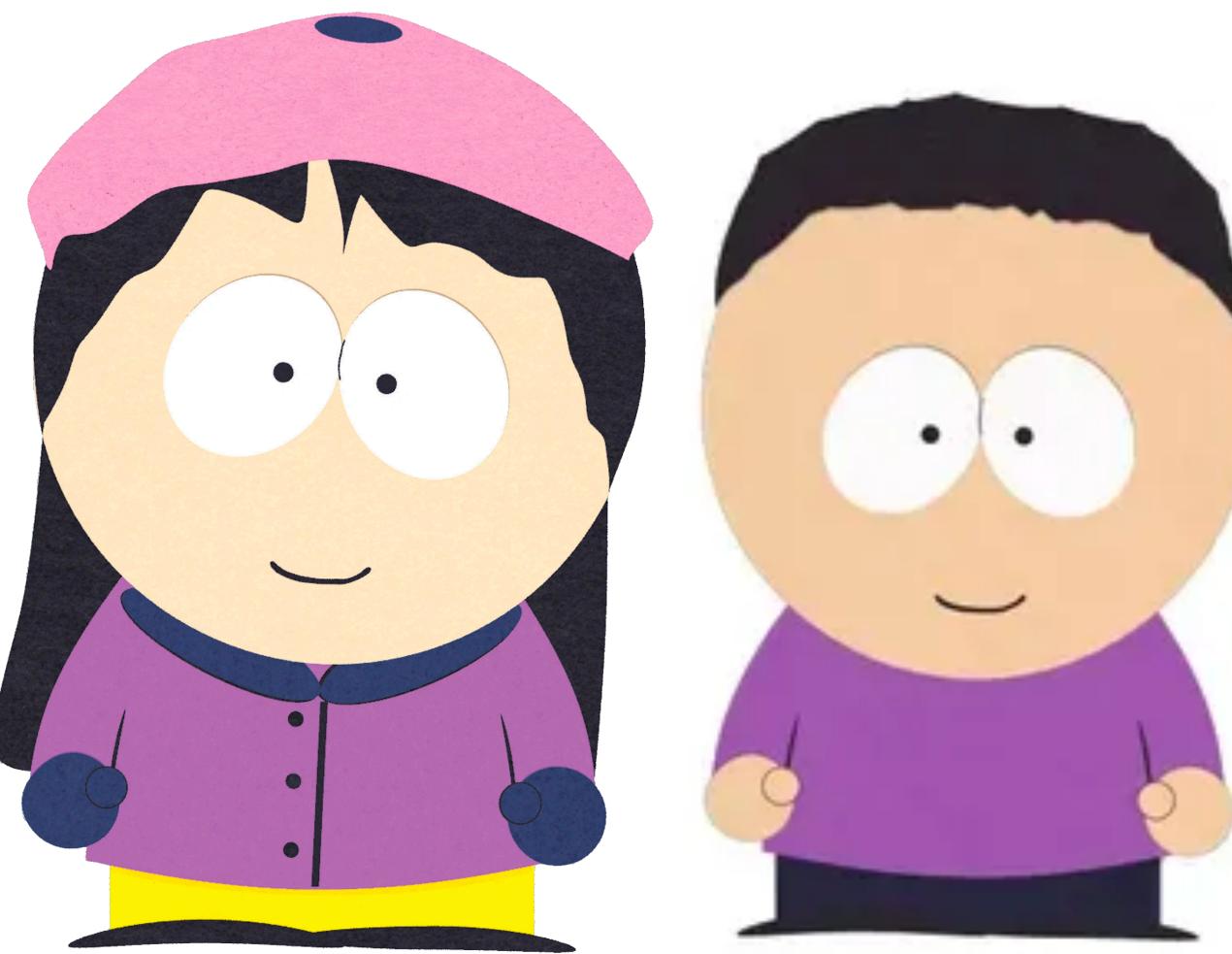


Все шансы есть
подготовить хороший
доклад!

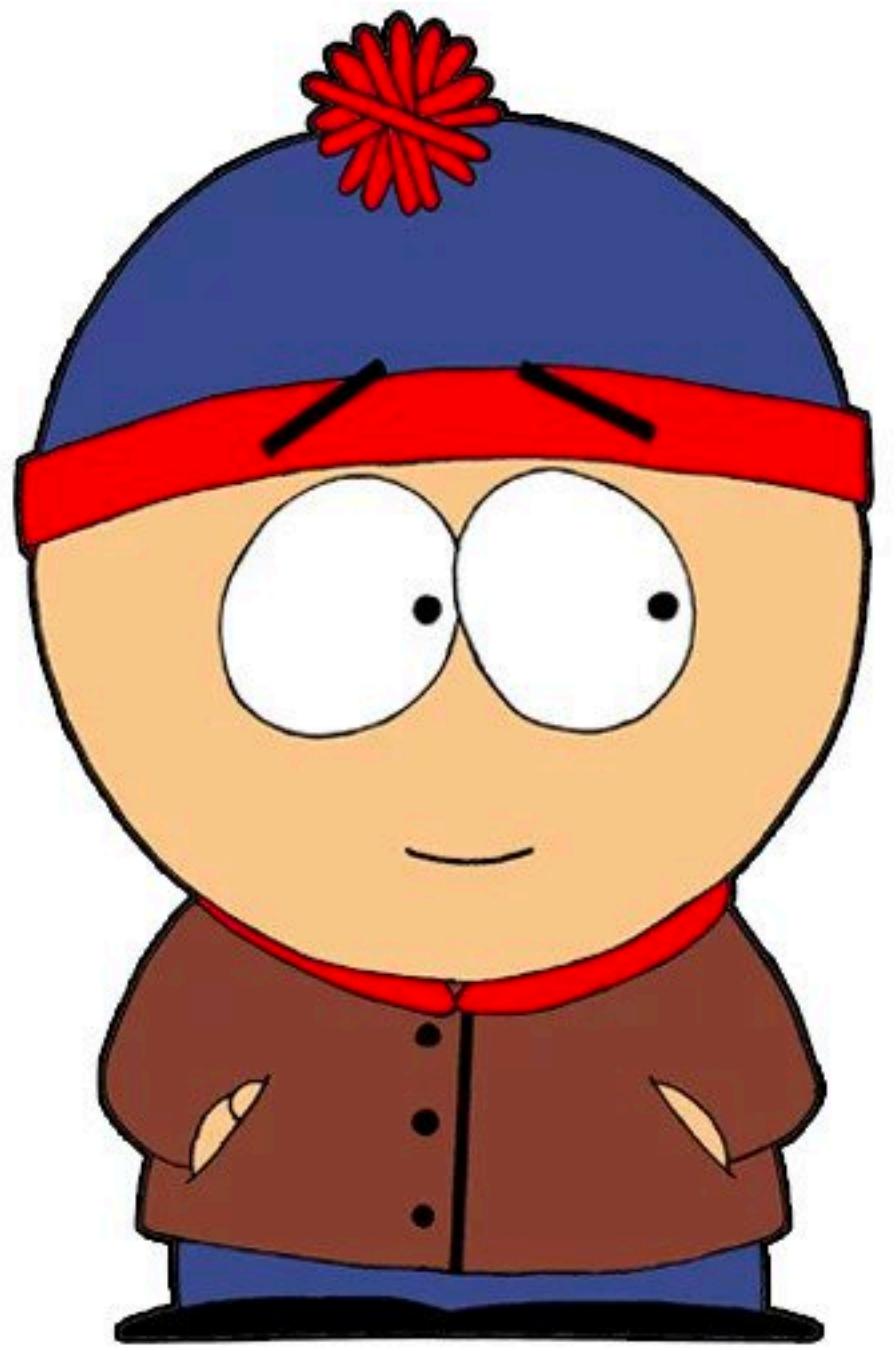




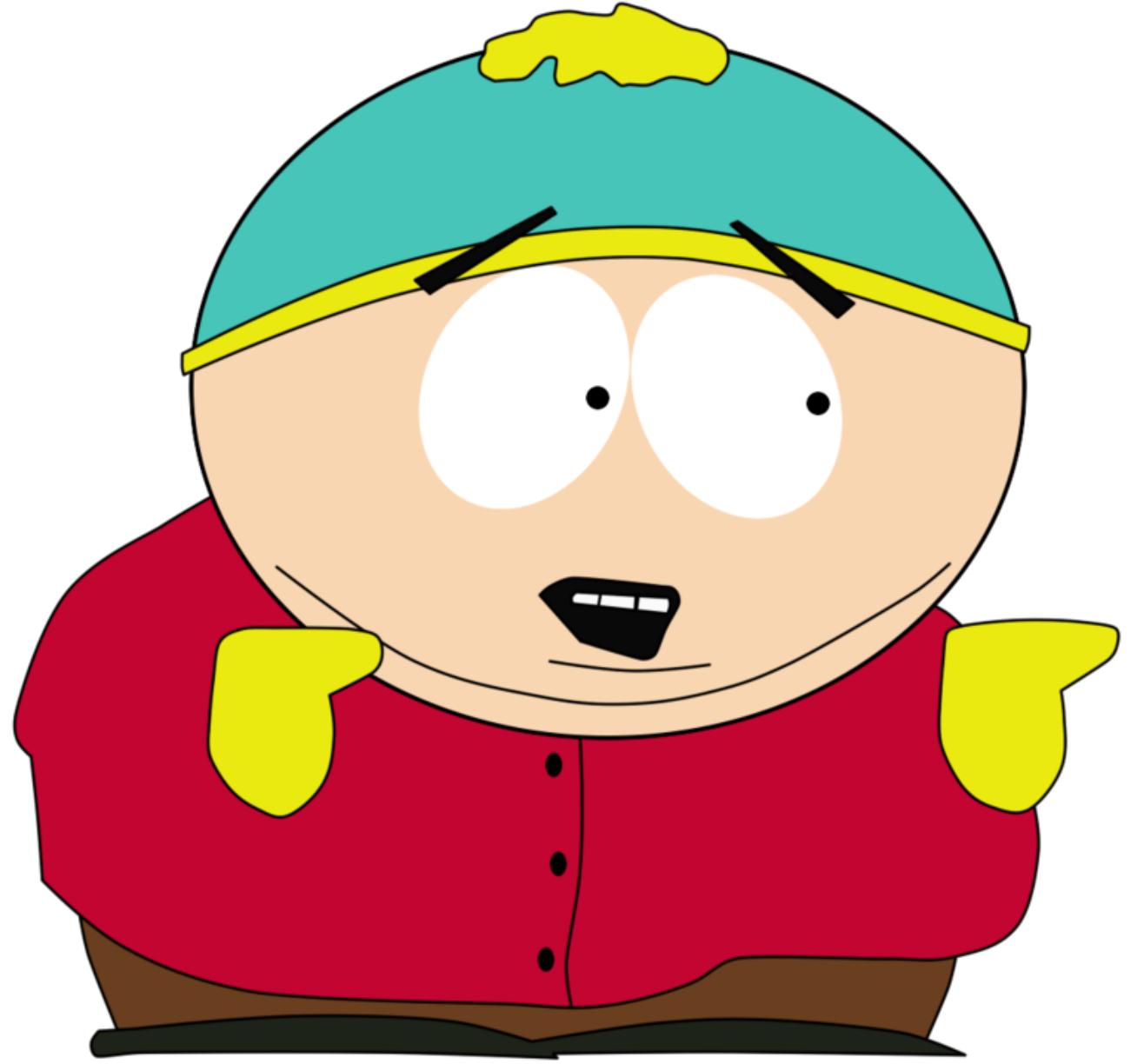
Но может не успеешь
подготовить, но можно
пробовать в запас.



В следующий раз буду готовиться заранее



Набросал детальный план доклада





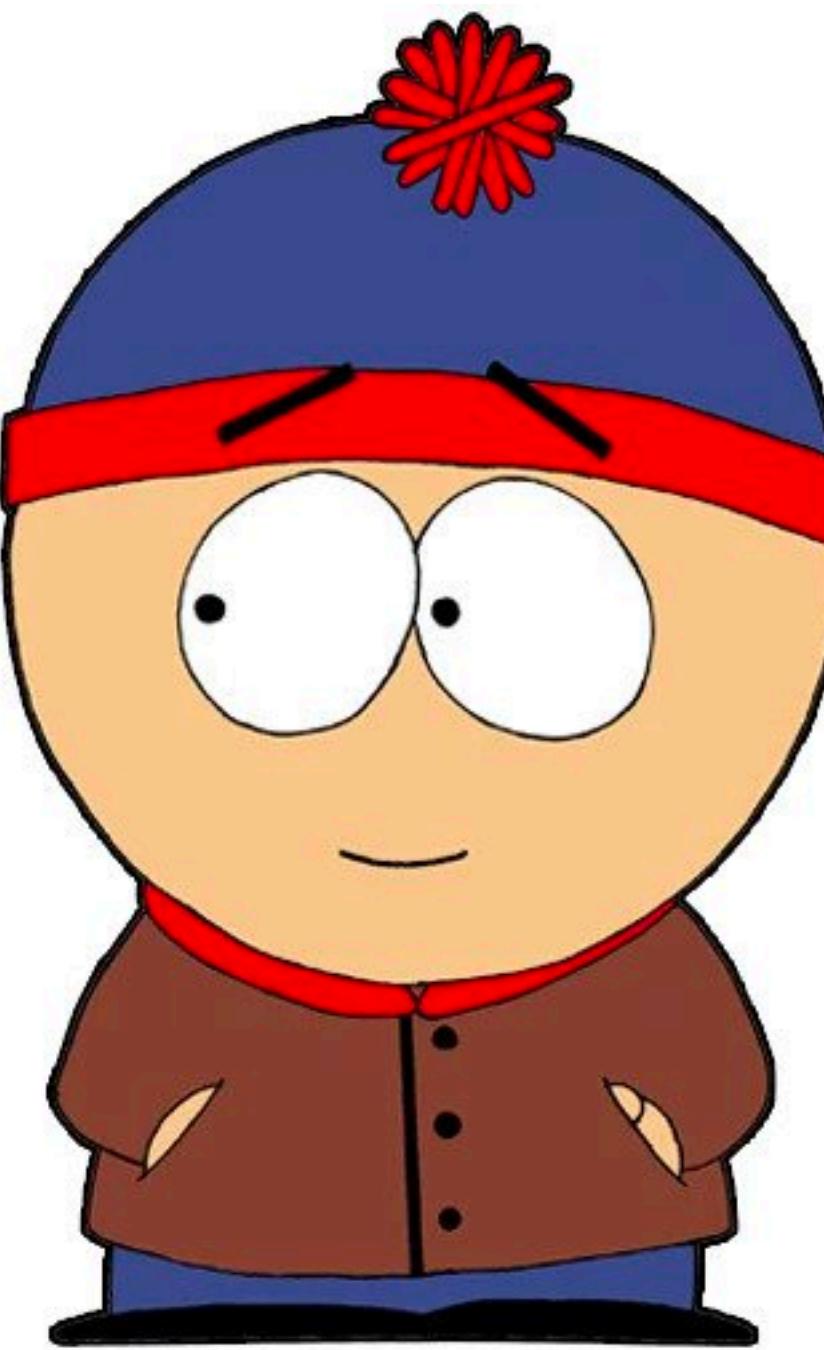
Доклад отличный!



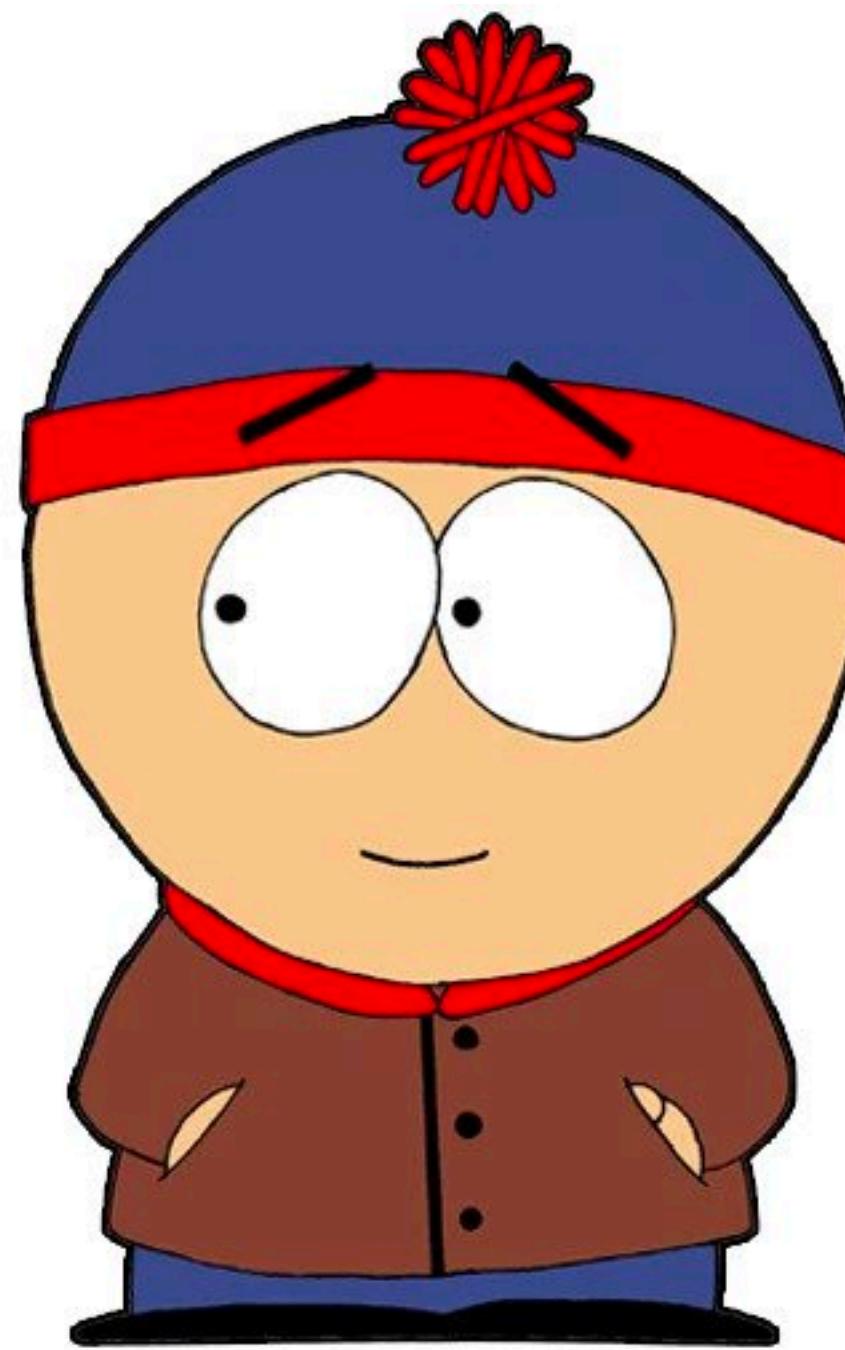
Доклад отличный!
Но нужно все
переделать!



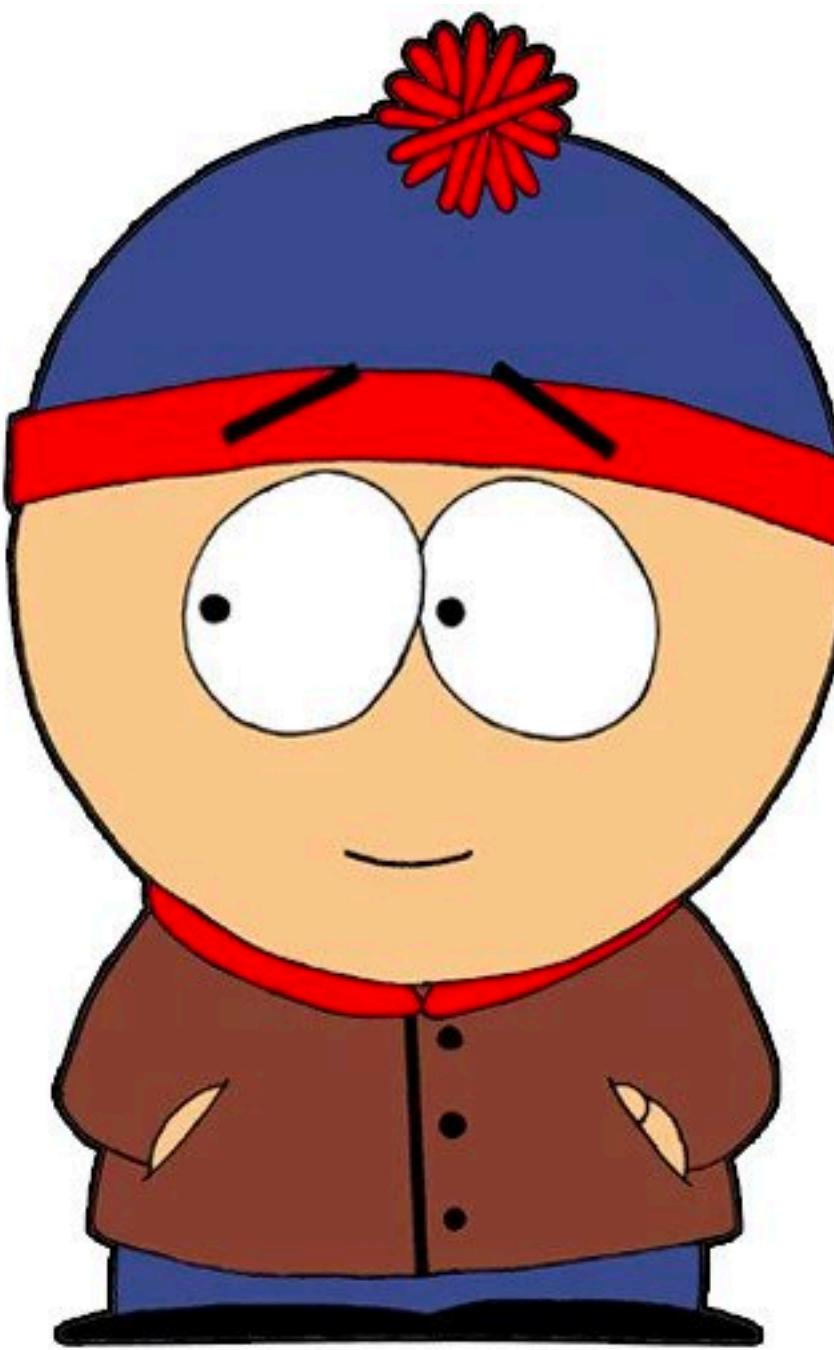
Слайды нужно
переделывать



Дополнить контентом



Оказывается, что мне
очевидно — всем совсем
не очевидно



**А помните на слайде с
изображением Картмана...**



Нет, не на этом, а на другом.

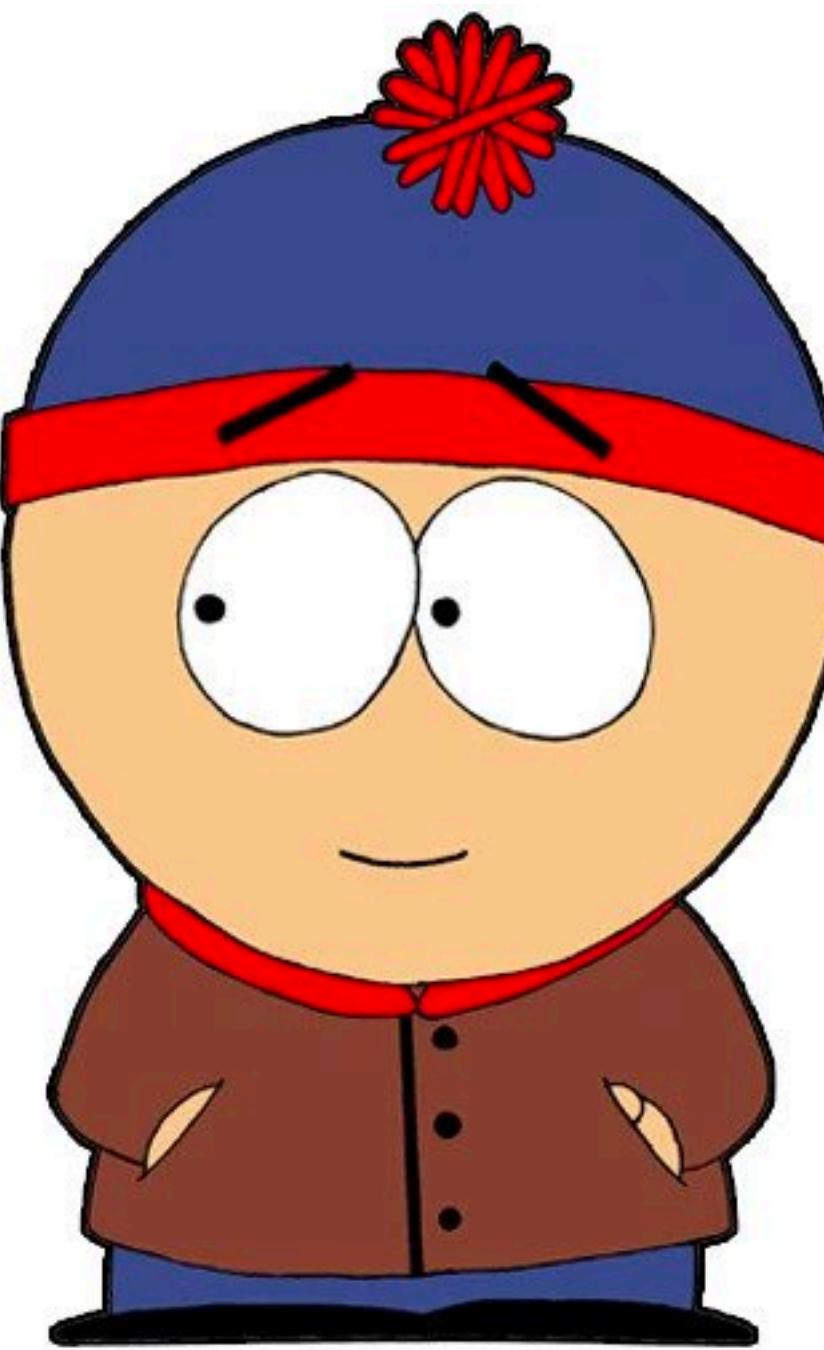


На другом!





**Слайды нужно
нумеровать...**





Вот этот слайд с кодом...

```

public static bool IsMultiSigContract(this byte[] script)
{
    int m, n = 0;
    int i = 0;
    if (script.Length < 37) return false;
    if (script[i] > (byte)OpCode.PUSH16) return false;
    if (script[i] < (byte)OpCode.PUSH1 && script[i] != 1 && script[i] != 2) return false;
    switch (script[i])
    {
        case 1:
            m = script[++i];
            ++i;
            break;
        case 2:
            m = script.ToInt16(++i);
            i += 2;
            break;
        default:
            m = script[i++]-80;
            break;
    }
    if (m < 1 || m > 1024) return false;
    while (script[i] == 33)
    {
        i += 34;
        if (script.Length <= i) return false;
        ++n;
    }
    if (n < m || n > 1024) return false;
    switch (script[i])
    {
        case 1:
            if (n != script[++i]) return false;
            ++i;
            break;
        case 2:
            if (script.Length < i + 3 || n != script.ToInt16(++i)) return false;
            i += 2;
            break;
        default:
            if (n != script[i++]-80) return false;
            break;
    }
    if (script[i++] != (byte)OpCode.CHECKMULTISIG) return false;
    if (script.Length != i) return false;
    return true;
}

```

```

switch (script[i])
{
    case 1:
        if (n != script[+i]) return false;
        ++i;
        break;
    case 2:
        if (script.Length < i + 3 || n != script.ToInt16(+i)) return false;
        i += 2;
        break;
    default:
        if (n != script[i++]-80) return false;
        break;
}

```



```

public static bool IsSignatureContract(this byte[] script)
{
    if (script.Length != 35) return false;
    if (script[0] != 33 || script[34] != (byte)OpCode.CHECKSIG)
        return false;
    return true;
}

```

```

public static bool IsMultiSigContract(this byte[] script)
{
    int m, n = 0;
    int i = 0;
    if (script.Length < 37) return false;
    if (script[i] > (byte)OpCode.PUSH16) return false;
    if (script[i] < (byte)OpCode.PUSH1 && script[i] != 1 && script[i] != 2) return false;
    switch (script[i])
    {
        case 1:
            m = script[++i];
            ++i;
            break;
        case 2:
            m = script.ToInt16(++i);
            i += 2;
            break;
        default:
            m = script[i++]-80;
            break;
    }
    if (m < 1 || m > 1024) return false;
    while (script[i] == 33)
    {
        i += 34;
        if (script.Length <= i) return false;
        ++n;
    }
    if (n < m || n > 1024) return false;
    switch (script[i])
    {
        case 1:
            if (n != script[++i]) return false;
            ++i;
            break;
        case 2:
            if (script.Length < i + 3 || n != script.ToInt16(++i)) return false;
            i += 2;
            break;
        default:
            if (n != script[i++]-80) return false;
            break;
    }
    if (script[i++] != (byte)OpCode.CHECKMULTISIG) return false;
    if (script.Length != i) return false;
    return true;
}

```

```

switch (script[i])
{
    case 1:
        if (n != script[+i]) return false;
        ++i;
        break;
    case 2:
        if (script.Length < i + 3 || n != script.ToInt16(+i)) return false;
        i += 2;
        break;
    default:
        if (n != script[i++]-80) return false;
        break;
}

```



```

public static bool IsSignatureContract(this byte[] script)
{
    if (script.Length != 35) return false;
    if (script[0] != 33 || script[34] != (byte)OpCode.CHECKSIG)
        return false;
    return true;
}

```

```

public static bool IsMultiSigContract(this byte[] script)
{
    int m, n = 0;
    int i = 0;
    if (script.Length < 37) return false;
    if (script[i] > (byte)OpCode.PUSH16) return false;
    if (script[i] < (byte)OpCode.PUSH1 && script[i] != 1 && script[i] != 2) return false;
    switch (script[i])
    {
        case 1:
            m = script[++i];
            ++i;
            break;
        case 2:
            m = script.ToInt16(++i);
            i += 2;
            break;
        default:
            m = script[i++]-80;
            break;
    }
    if (m < 1 || m > 1024) return false;
    while (script[i] == 33)
    {
        i += 34;
        if (script.Length <= i) return false;
        ++n;
    }
    if (n < m || n > 1024) return false;
    switch (script[i])
    {
        case 1:
            if (n != script[++i]) return false;
            ++i;
            break;
        case 2:
            if (script.Length < i + 3 || n != script.ToInt16(++i)) return false;
            i += 2;
            break;
        default:
            if (n != script[i++]-80) return false;
            break;
    }
    if (script[i++] != (byte)OpCode.CHECKMULTISIG) return false;
    if (script.Length != i) return false;
    return true;
}

```

```

switch (script[i])
{
    case 1:
        if (n != script[+i]) return false;
        ++i;
        break;
    case 2:
        if (script.Length < i + 3 || n != script.ToInt16(+i)) return false;
        i += 2;
        break;
    default:
        if (n != script[i++]-80) return false;
        break;
}

```



```

public static bool IsSignatureContract(this byte[] script)
{
    if (script.Length != 35) return false;
    if (script[0] != 33 || script[34] != (byte)OpCode.CHECKSIG)
        return false;
    return true;
}

```



**На нем плохо видно ...
И нужно переделать
форматирование!**



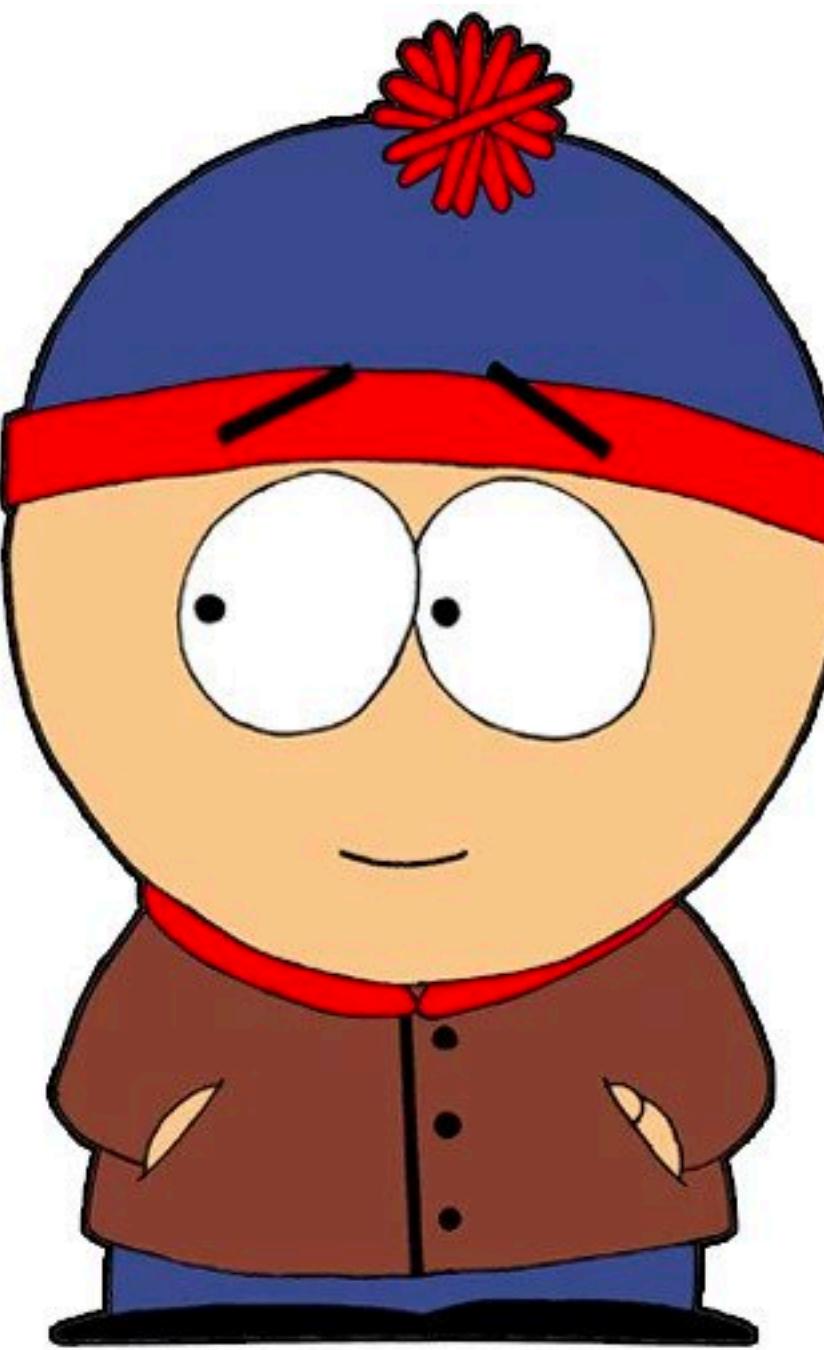


Переделал слайды как надо,
пойду на следующий прогон.

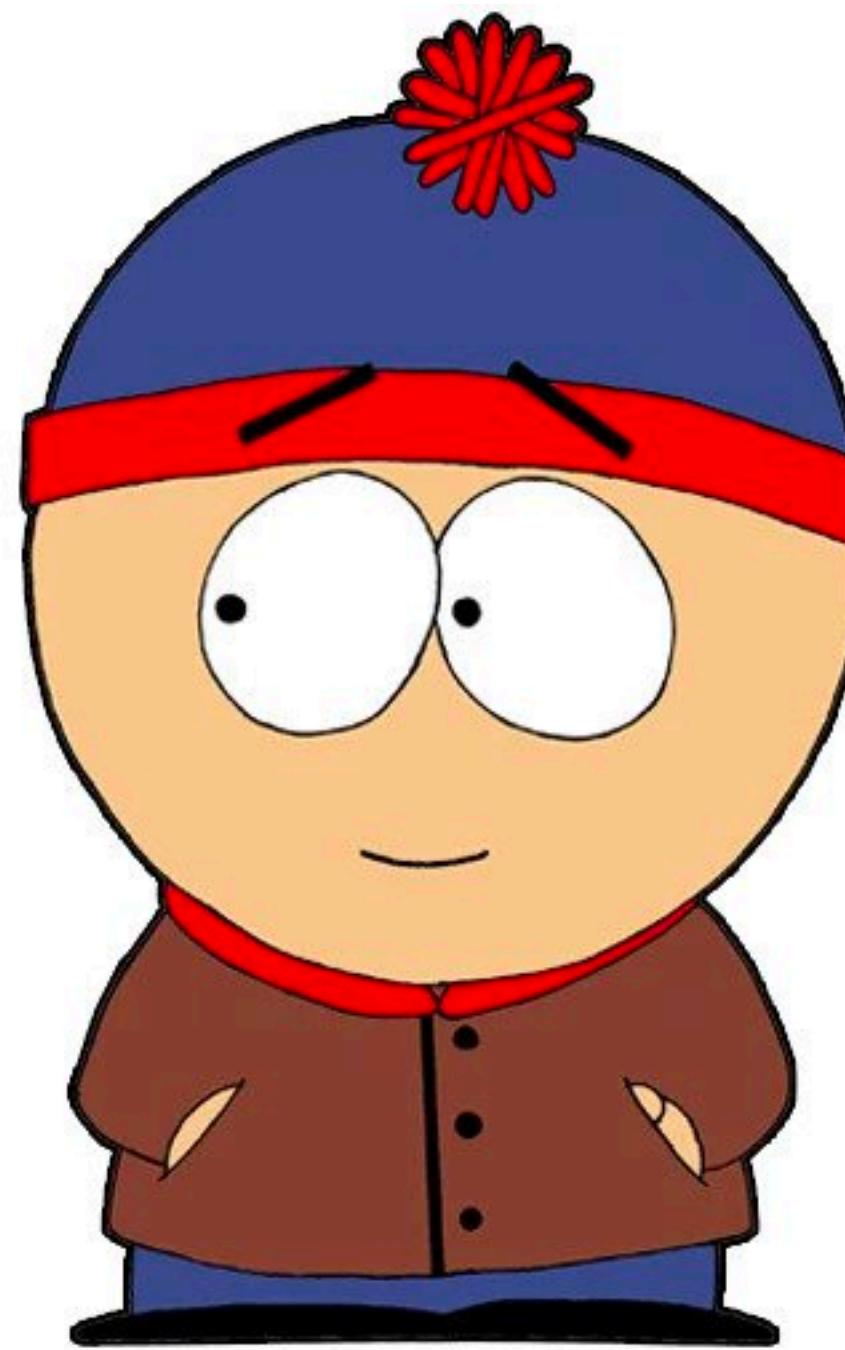


Слайды хорошие, но речь не
сходится с тем, что на слайде.
И переходов между слайдами
нет.

Еще раз буду делать
репетиции.



**Еще раз буду делать
репетиции.
С каждой — все лучше и
лучше!**





Интеренет отключился, когда в
4й раз показывал демо...

Записал видео с демо.



- Понять свою мотивацию (зачем мне делать доклад?)
- Подавать доклад не в последний момент
- Если даже есть интересная идея, но непонятно как правильно выстроить план доклада, то можно приходить
- Детальный план сильно помогает ПК понять, о чем я хочу рассказать, и помочь
- Оформление слайдов — это немаловажно
- Репетиций мало не бывает
- Доклад != лекция

Выводов бы поменьше



- Алексей Каптерев “Мастерство презентации”
- <https://bit.ly/2rYXxxm>
- <https://wiki.jugru.org/display/SPEAKERS/Memo+for+speakers#Memoforspeakers-Sourcecode>

HEISENBUG

Тестирование. Не только для тестировщиков!

Подать доклад



8-9 апреля 2020

Санкт-Петербург, Петербургское шоссе 64к1 лит. А



ЭКСПОФОРУМ

Спасибо!