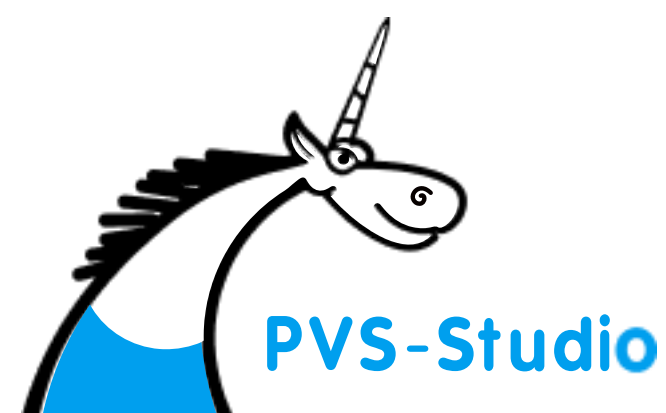
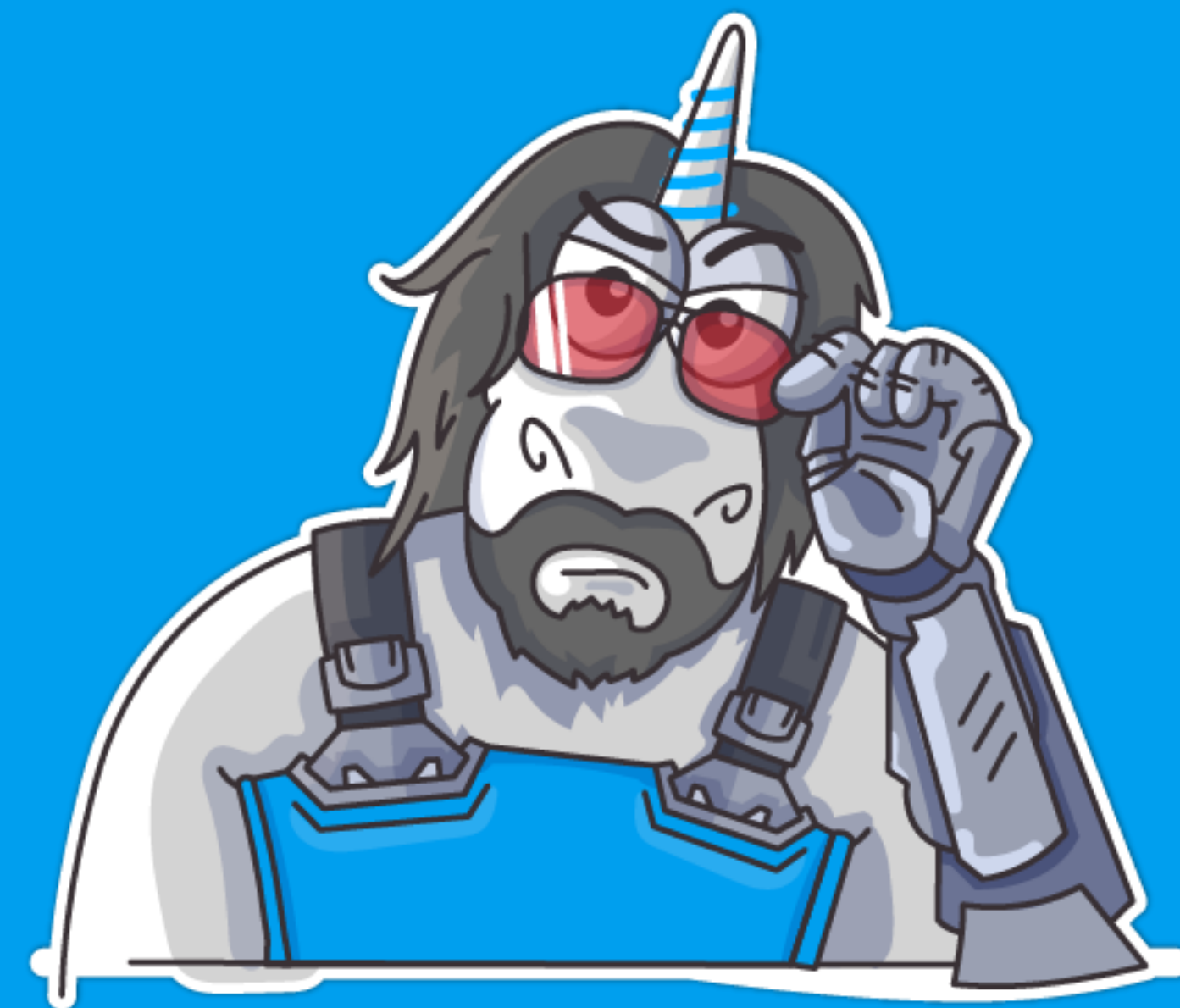


Статический анализ vs GameDev

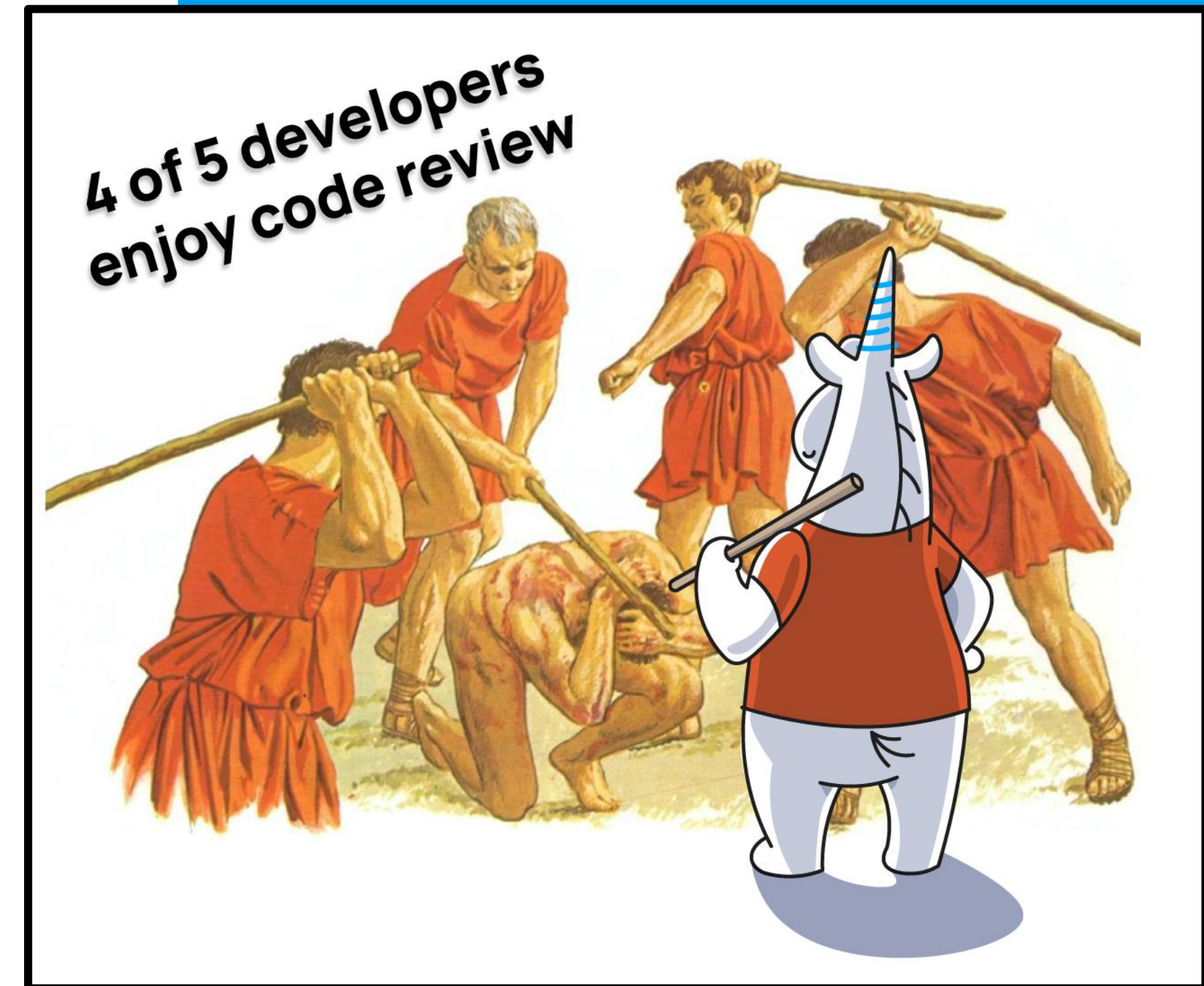
Вебинар «Оптимизация игр»
25.09.25



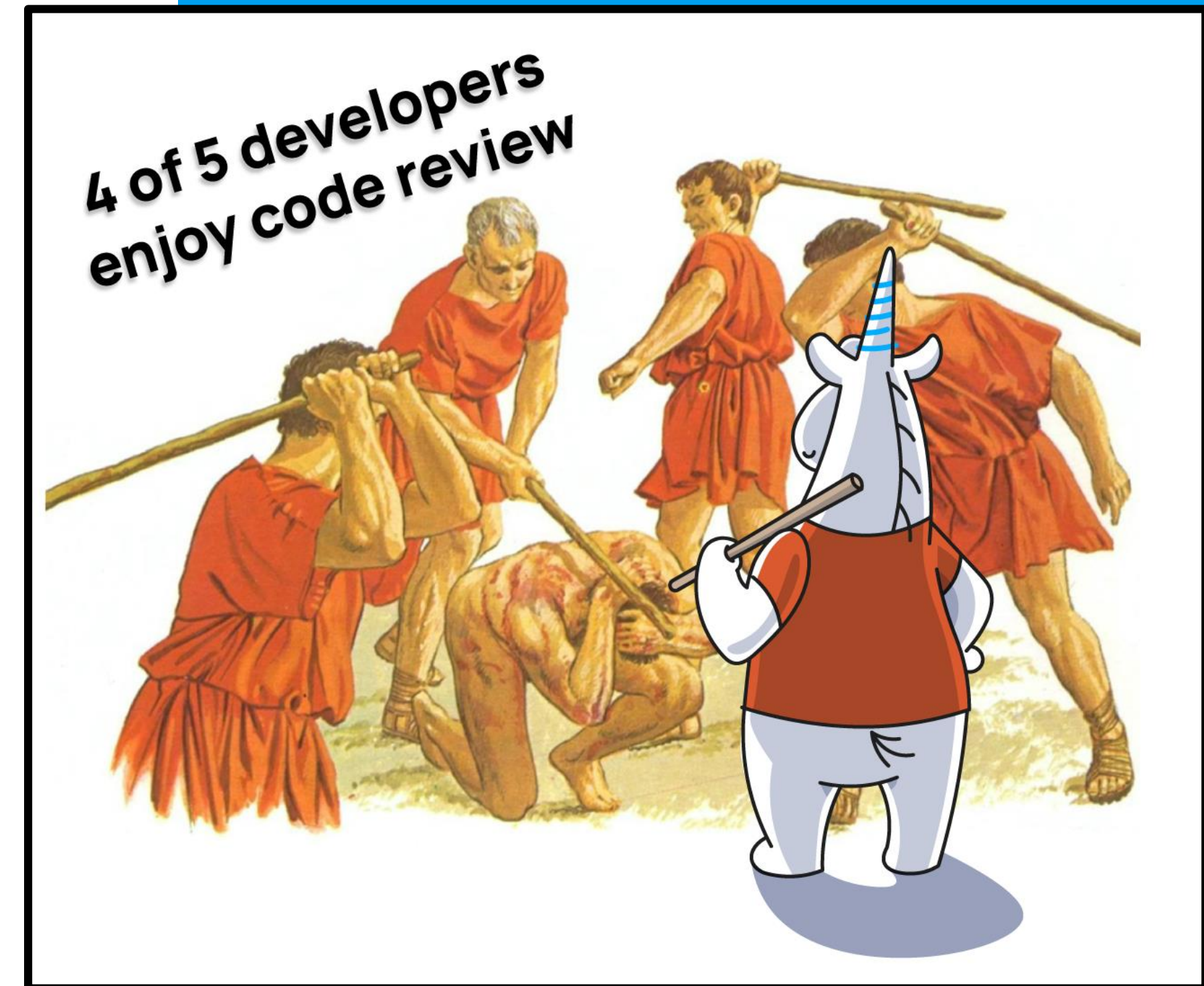
Глеб Асламов
C# Developer Advocate



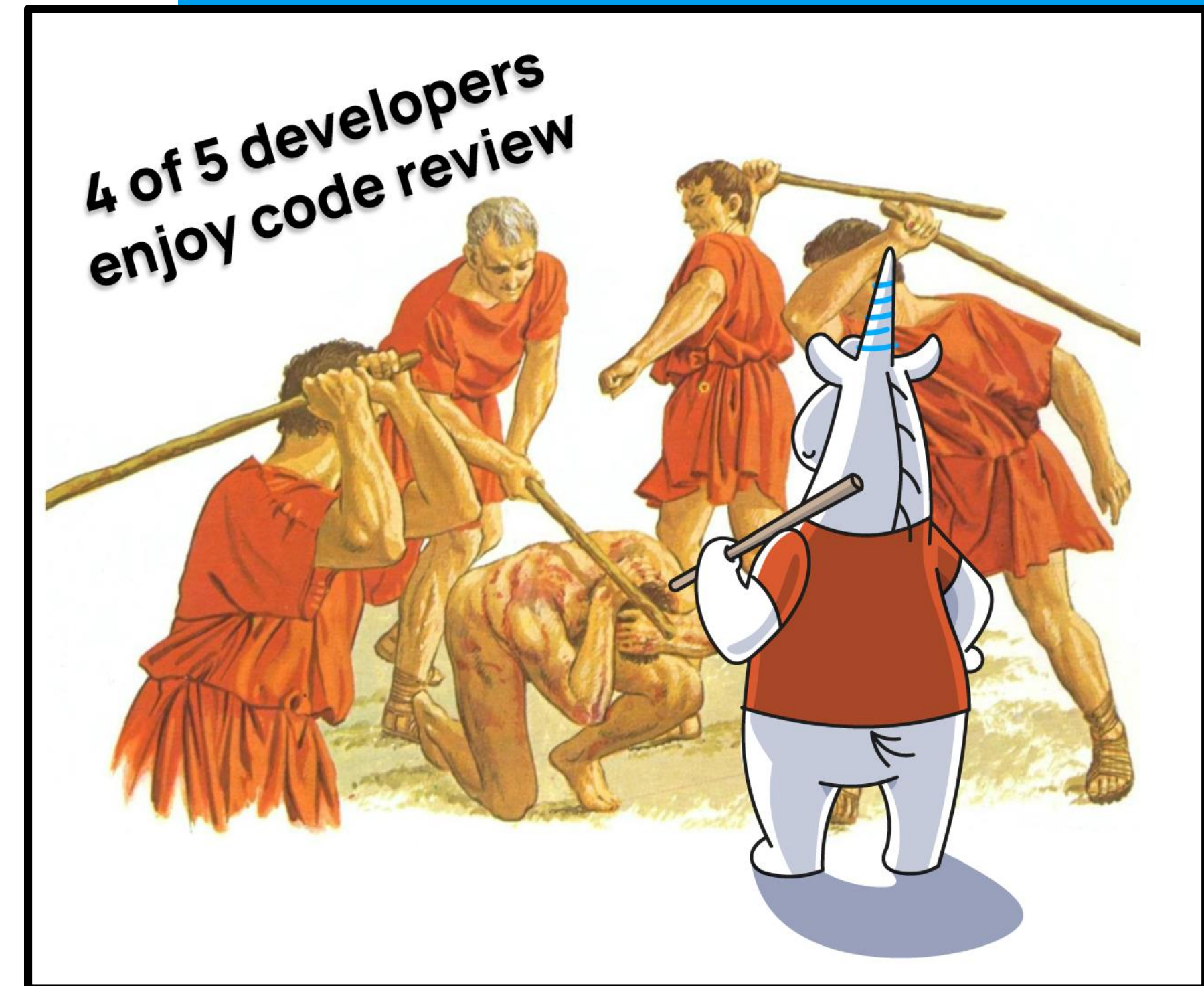
- В начале был код-ревью...



- Автоматический код-ревью!



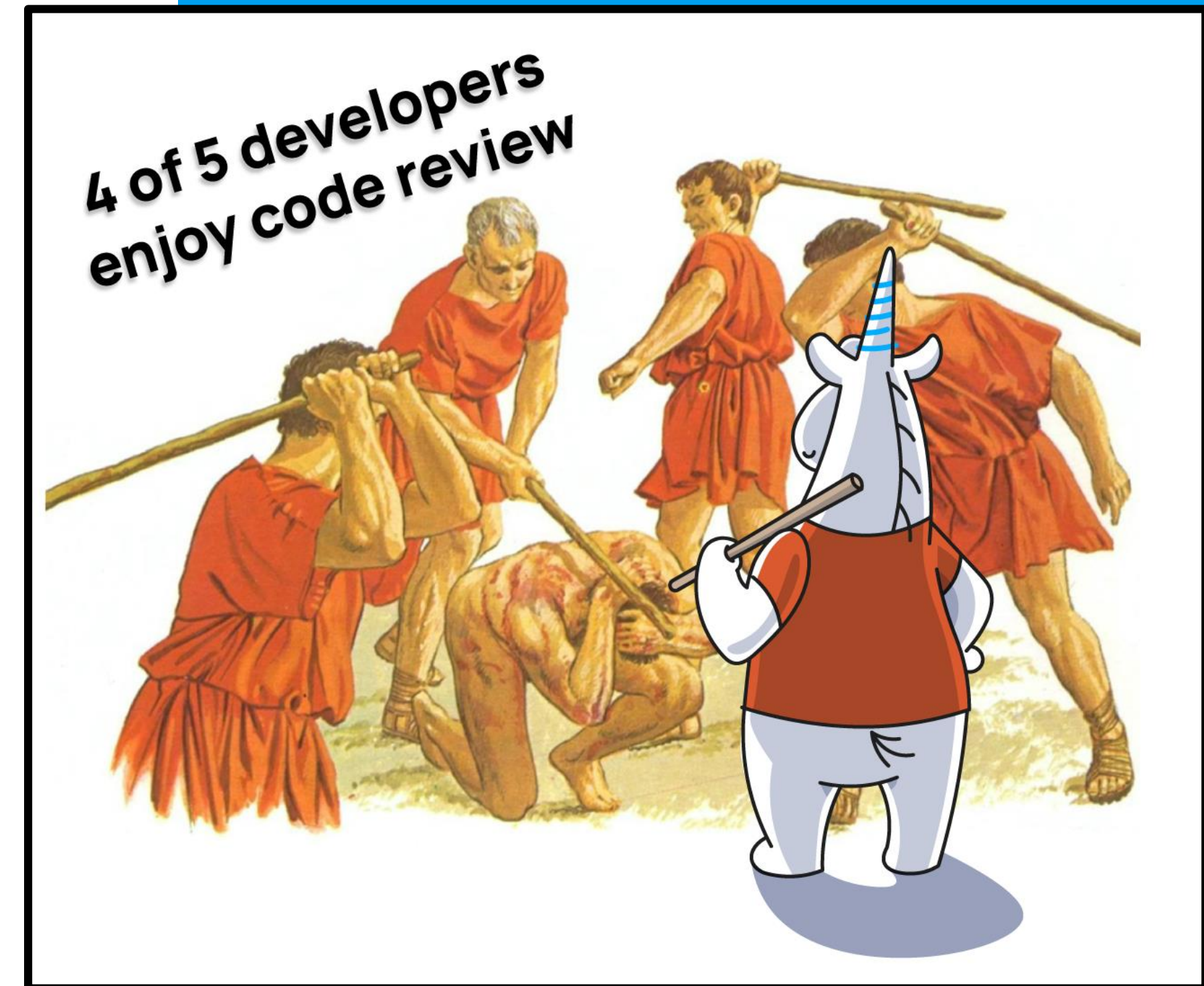
- Автоматический код-ревью!
- Нужен только код



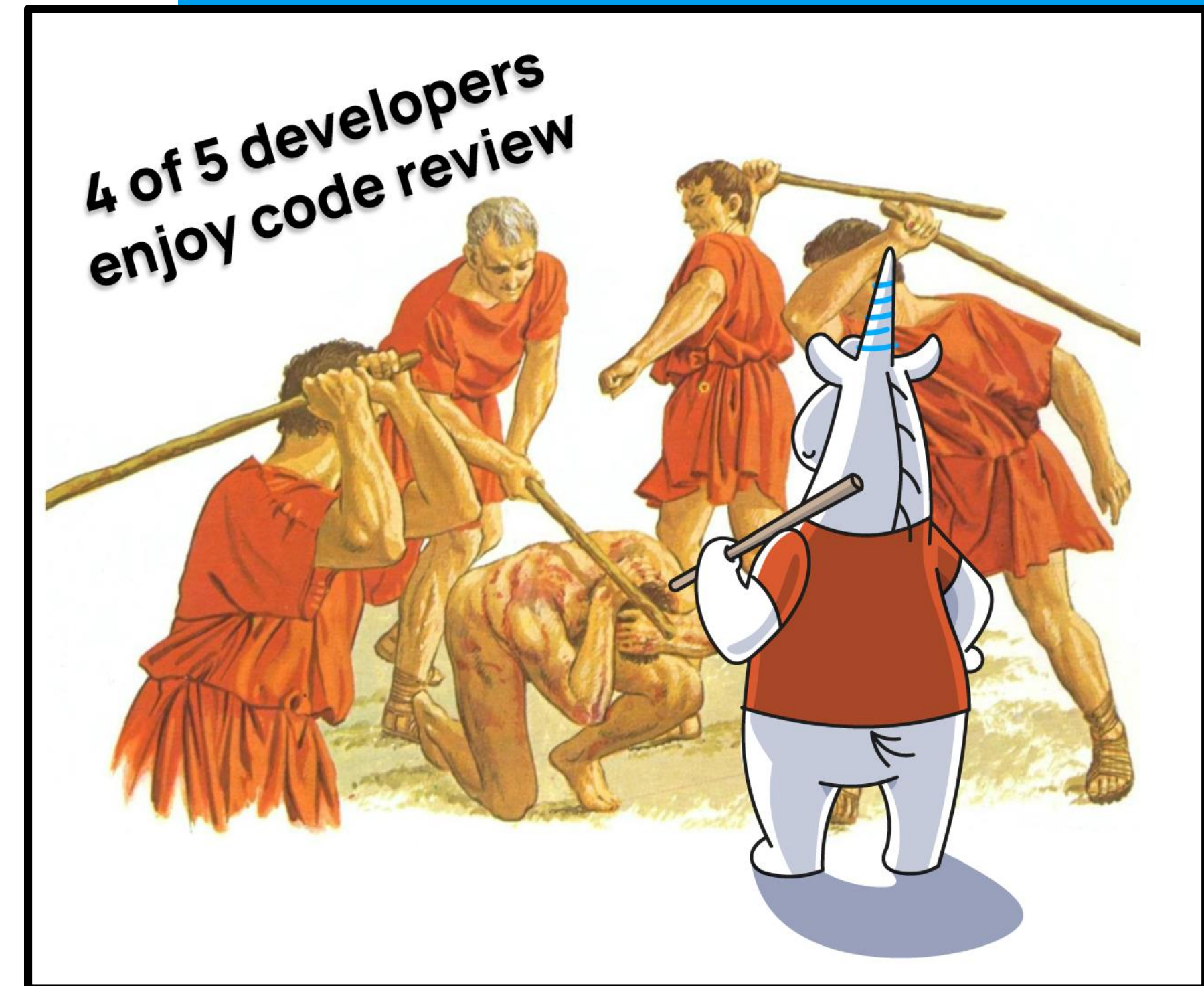
Статический анализ

6

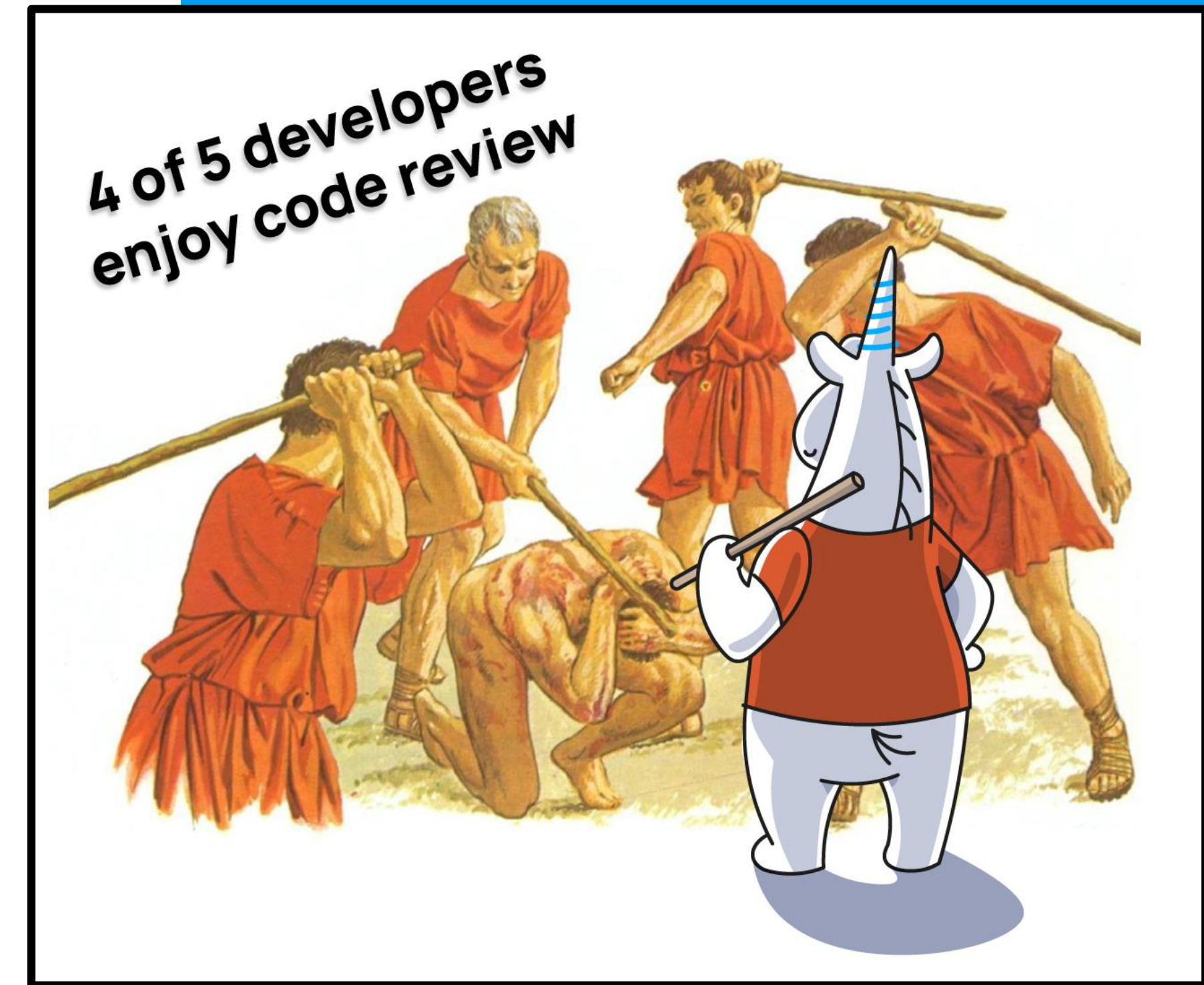
- Автоматический код-ревью!
- Нужен только код
- Полное покрытие



- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок



- Автоматический код-ревью!
- Нужен только код
- Полное покрытие
- Раннее обнаружение ошибок
- Ошибки исправляются на этапе разработки



Виды проблем

проблемы
безопасности

неправильная работа
с методами

недостижимый
код

ошибки доступа к
памяти

опечатки



ошибки сериализации /
десериализации

выход за
границы

ошибки
синхронизации

переполнение
буфера

неправильная работа с
типами

Что находит PVS-Studio в реальности?




```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        ExpiryTime == other.ExpiryTime &&  
        Identifier == other.Identifier &&  
        IPRange == other.IPRange &&  
        Permissions == other.Permissions &&  
        Protocol == other.Protocol &&  
        StartTime == other.StartTime &&  
        Version == other.Version;  
}
```

```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        ExpiryTime == other.ExpiryTime &&  
        Identifier == other.Identifier &&  
        IPRange == other.IPRange &&  
        Permissions == other.Permissions &&  
        Protocol == other.Protocol &&  
        StartTime == other.StartTime &&  
        Version == other.Version;  
}
```



```
public struct BlobSasBuilder : IEquatable<BlobSasBuilder> {  
    ....  
    public bool Equals(BlobSasBuilder other) =>  
        BlobName == other.BlobName &&  
        CacheControl == other.CacheControl &&  
        BlobContainerName == other.BlobContainerName &&  
        ContentDisposition == other.ContentDisposition &&  
        ContentEncoding == other.ContentEncoding &&  
        ContentLanguage == other.ContentEncoding &&  
        ContentType == other.ContentType &&  
        ExpiryTime == other.ExpiryTime &&  
        Identifier == other.Identifier &&  
        IPRange == other.IPRange &&  
        Permissions == other.Permissions &&  
}
```

Предупреждение PVS-Studio:

V3112 An abnormality within similar comparisons. It is possible that a typo is present inside the expression '`ContentLanguage == other.ContentEncoding`'

```
public struct FileSasBuilder : IEquatable<FileSasBuilder> {  
    ....  
    public bool Equals(FileSasBuilder other) =>  
        CacheControl == other.CacheControl  
        && ContentDisposition == other.ContentDisposition  
        && ContentEncoding == other.ContentEncoding  
        && ContentLanguage == other.ContentEncoding  
        && ContentType == other.ContentType  
        && ExpiryTime == other.ExpiryTime  
        && FilePath == other.FilePath  
        && Identifier == other.Identifier  
        && IPRange == other.IPRange  
        && Permissions == other.Permissions  
        && Protocol == other.Protocol  
        && ShareName == other.ShareName  
        && StartTime == other.StartTime  
        && Version == other.Version  
}
```



```
public struct FileSasBuilder : IEquatable<FileSasBuilder> {  
    ....  
    public bool Equals(FileSasBuilder other) =>  
        CacheControl == other.CacheControl  
        && ContentDisposition == other.ContentDisposition  
        && ContentEncoding == other.ContentEncoding  
        && ContentLanguage == other.ContentEncoding  
        && ContentType == other.ContentType  
        && ExpiryTime == other.ExpiryTime  
        && FilePath == other.FilePath  
        && Identifier == other.Identifier  
        && IPRange == other.IPRange  
        && Permissions == other.Permissions  
        && Protocol == other.Protocol  
        && ShareName == other.ShareName  
        && StartTime == other.StartTime  
        && Version == other.Version  
}
```

Анализ потока данных [Data Flow]

16

- Определяет предположительное значение переменных и констант
- Примеры артефактов:
 - диапазон значений
 - точное значение
 - множество значений

```
int number = random.Next(1, 10)  
if (number == 10) ←
```



PVS-Studio


```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        . . .
    }
}
```

```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        if ( expr.Arguments.Count != 1)
        {
            throw new InvalidOperationException("...");
        }
        ...
    }
}
```

```
public override void VisitMethod(MethodExpression expr)
{
    if (    expr.Name.Value == "id"
        && expr.Arguments.Count == 0)
    {
        if ( expr.Arguments.Count != 1)
        {
            throw new InvalidOperationException("...");
        }
        . . .
    }
}
```

Предупреждение PVS-Studio:

V3022 Expression 'expr.Arguments.Count != 1' is always true.

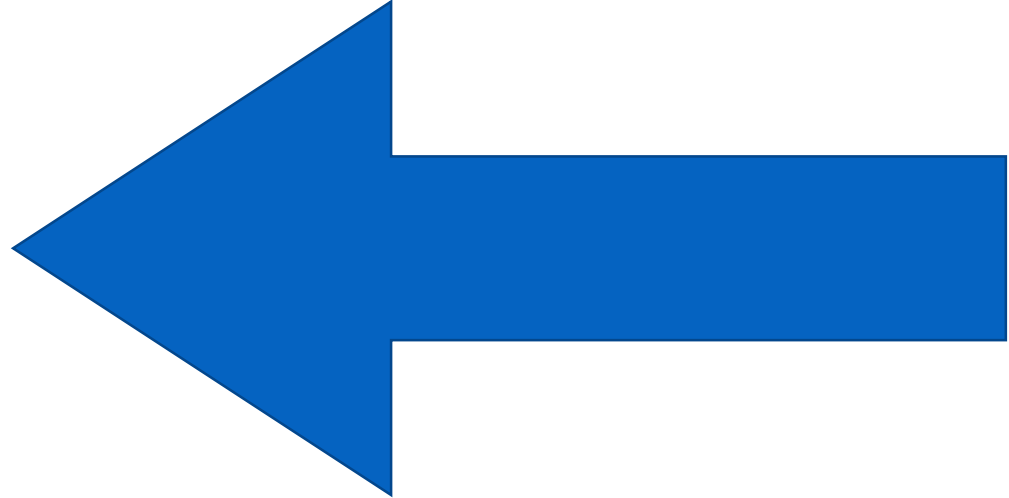
А где GameDev?



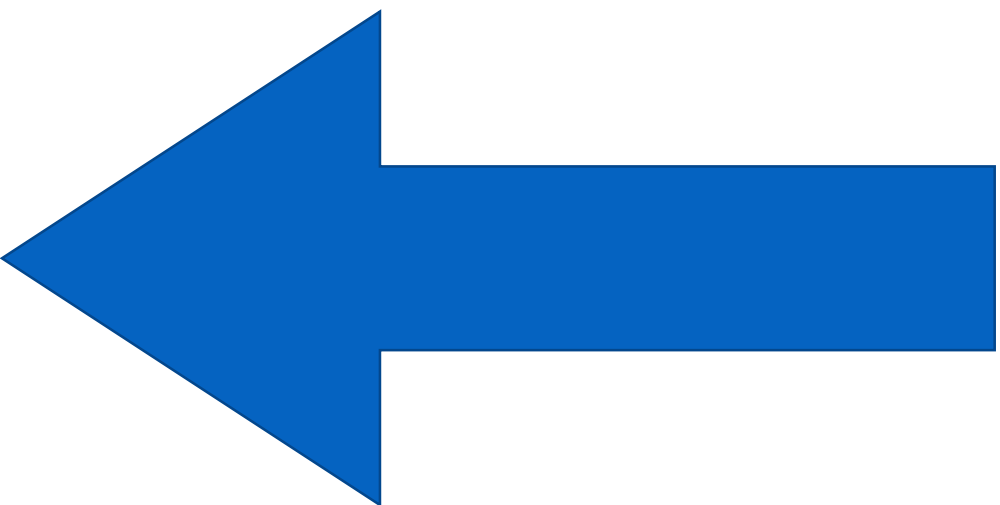
Межмодульный анализ [Barotrauma]

21

```
// Функция Remove() делает ссылку Sprite нулевой
partial class DecorativeSprite : ISerializableEntity
{
    public Sprite Sprite { get; private set; }
    . . .
    public void Remove()
    {
        Sprite?.Remove() ;
        Sprite = null;
        . . .
    }
}
```

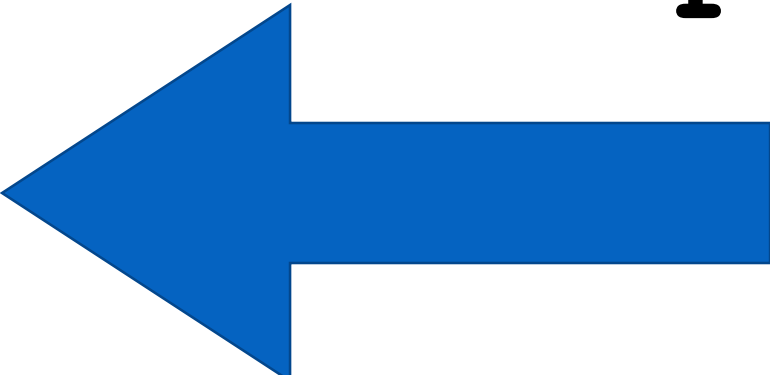


```
// Функция Remove() делает ссылку Sprite нулевой
partial class DecorativeSprite : ISerializableEntity
{
    public Sprite Sprite { get; private set; }
    . . .
    public void Remove()
    {
        Sprite?.Remove();
        Sprite = null;
        . . .
    }
}
```

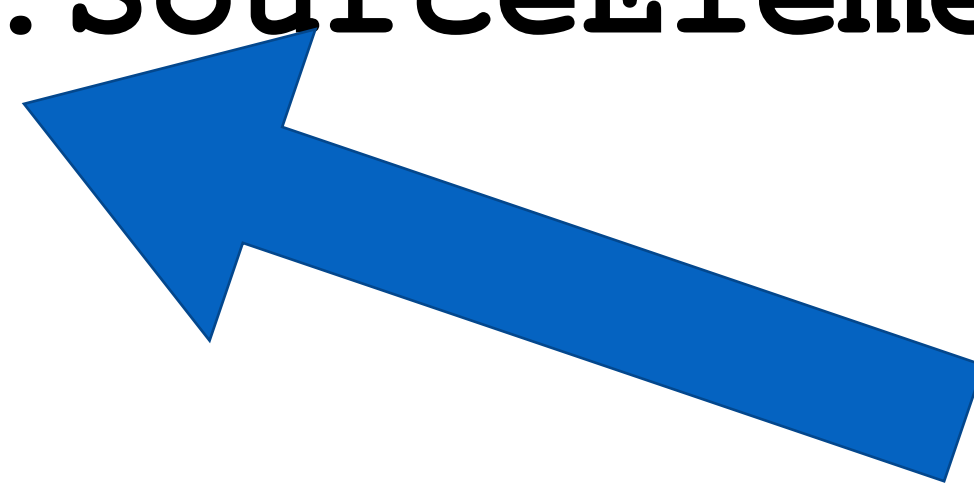



```
public void RecreateSprites()  
{  
    for (int i = 0; i < DecorativeSprites.Count; i++)  
    {  
        var decorativeSprite = DecorativeSprites[i];  
        decorativeSprite.Remove();  
        var source =  
            decorativeSprite.Sprite.SourceElement;  
        . . .  
    }  
}
```

```
public void RecreateSprites()  
{  
    for (int i = 0; i < DecorativeSprites.Count; i++)  
    {  
        var decorativeSprite = DecorativeSprites[i];  
        decorativeSprite.Remove();  
        var source =  
            decorativeSprite.Sprite.SourceElement;  
        . . .  
    }  
}
```



```
public void RecreateSprites()  
{  
    for (int i = 0; i < DecorativeSprites.Count; i++)  
    {  
        var decorativeSprite = DecorativeSprites[i];  
        decorativeSprite.Remove();  
        var source =  
            decorativeSprite.Sprite.SourceElement;  
        . . .  
    }  
}
```



Предупреждение PVS-Studio:

V3080. Possible null dereference. Consider inspecting 'decorativeSprite.Sprite'.


```
void RecvProxy_QuaternionToQuaternion(  
    const CRecvProxyData *pData,  
    void *pStruct, void *pOut )  
{  
    const float *v = pData->m_Value.m_Vector;  
  
    Assert( IsFinite( v[0] ) && IsFinite( v[1] ) &&  
            IsFinite( v[2] ) && IsFinite( v[3] ) );  
    ((float*)pOut)[0] = v[0];  
    ((float*)pOut)[1] = v[1];  
    ((float*)pOut)[2] = v[2];  
    ((float*)pOut)[3] = v[3];  
}
```

Предупреждение PVS-Studio:

V557 Array overrun is possible. The '3' index is pointing beyond array bound.

```
void *pStruct, void *pOut )
{
    const float *v = pData->m_Value.m_Vector;

    Assert( IsFinite( v[0] ) && IsFinite( v[1] ) &&
            IsFinite( v[2] ) && IsFinite( v[3] ) );
    (float*)pOut[0] = v[0];
    (float*)pOut[1] = v[1];
    (float*)pOut[2] = v[2];
    (float*)pOut[3] = v[3];
}
```


```
class Dvariant {
    ....
    float m_Vector[3];
};

// This is passed into RecvProxy functions.
class CRecvProxyData {
    ....
    DVariant m_Value; // The value given to you to store.
};

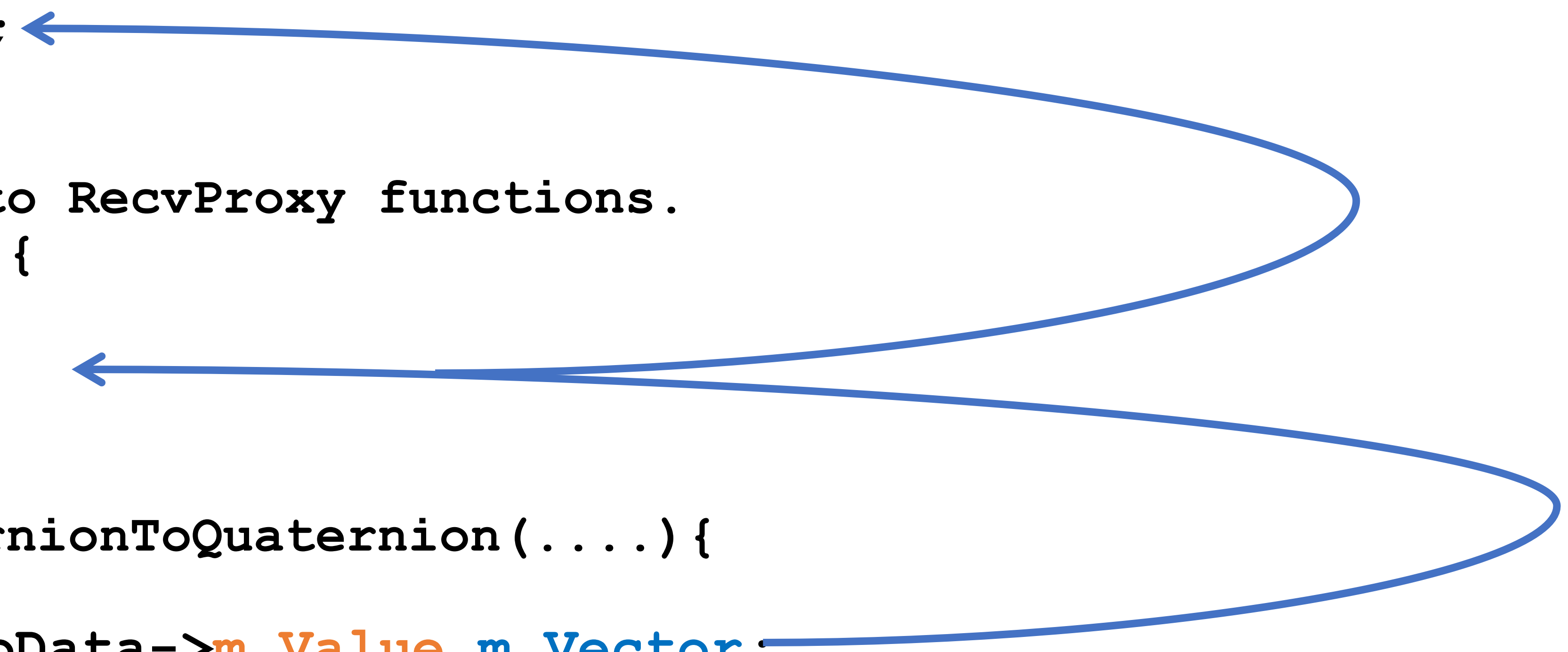
void RecvProxy_QuaternionToQuaternion(....) {
    ....
    const float *v = pData->m_Value.m_Vector;

    Assert (IsFinite( v[3] ));
    ((float*)pOut)[3] = v[3];
}
```



```
class Dvariant {  
    ....  
    float m_Vector[3];  
};  
  
// This is passed into RecvProxy functions.  
class CRecvProxyData {  
    ....  
    DVariant m_Value;   
};  
  
void RecvProxy_QuaternionToQuaternion(....) {  
    ....  
    const float *v = pData->m_Value.m_Vector;  
  
    Assert (IsFinite( v[3] ));  
    ((float*)pOut)[3] = v[3];  
}
```

```
class Dvariant {  
    ....  
    float m_Vector[3];  
};  
  
// This is passed into RecvProxy functions.  
class CRecvProxyData {  
    ....  
    DVariant m_Value;  
};  
  
void RecvProxy_QuaternionToQuaternion(....) {  
    ....  
    const float *v = pData->m_Value.m_Vector;  
  
    Assert (IsFinite( v[3] ));  
    ((float*)pOut)[3] = v[3];  
}
```



The diagram consists of three blue curved arrows indicating the flow of data. The first arrow starts at the `m_Vector` member of the `Dvariant` class and points to the `m_Value` member of the `CRecvProxyData` class. The second arrow starts at the `m_Value` member and points to the `v` pointer in the `RecvProxy_QuaternionToQuaternion` function. The third arrow starts at the `v` pointer and points to the `v[3]` array access in the `Assert` and assignment statements.

Ошибки при работе с массивами [Source SDK]

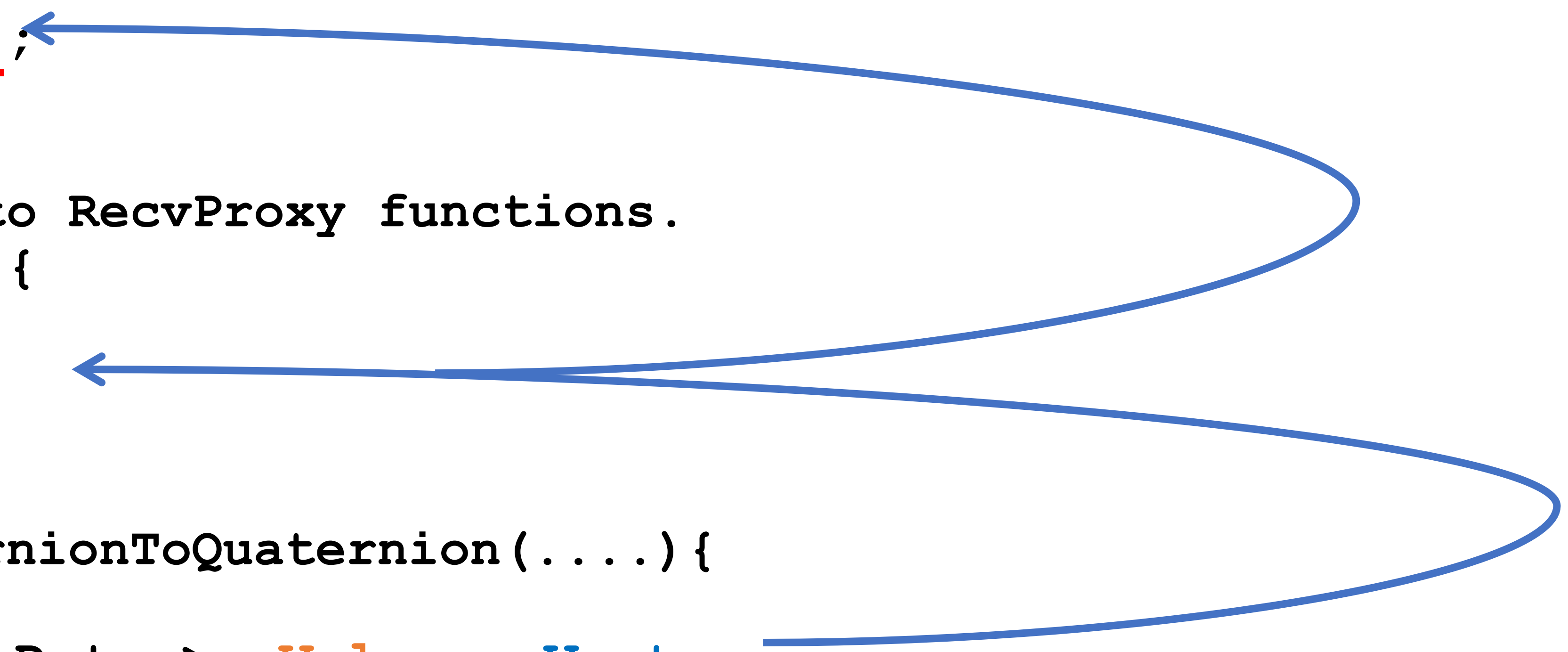
31

```
class Dvariant {
    ....
    float m_Vector[3];
};

// This is passed into RecvProxy functions.
class CRecvProxyData {
    ....
    DVariant m_Value;
};

void RecvProxy_QuaternionToQuaternion(....) {
    ....
    const float *v = pData->m_Value.m_Vector;

    Assert (IsFinite( v[3] ));
    ((float*)pOut)[3] = v[3];
}
```



The diagram consists of three blue curved arrows indicating the flow of data. The first arrow starts at the `m_Vector[3]` in the `Dvariant` class and points to the `m_Value` member of the `CRecvProxyData` class. The second arrow starts at the `m_Value` member and points to the `*v` pointer in the `RecvProxy_QuaternionToQuaternion` function. The third arrow starts at the `*v` pointer and points to the `v[3]` array access in the `Assert` and assignment statements.

```
union
{
    ....
    #if 0 // We can't ship this since it changes the size of DTVariant
        // to be 20 bytes instead of 16 and that breaks MODs!!!
        float m_Vector[4];
    #else
        float m_Vector[3];
    #endif
    ....
};
```



```
class SomeClass
{
    . . .
    UObject *m_ptr;
    . . .
};
```

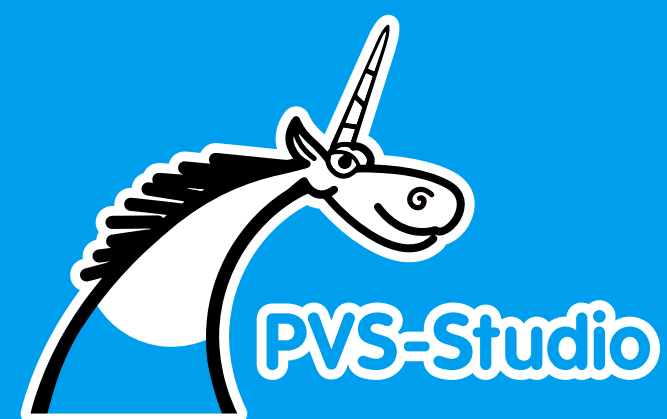
Предупреждение PVS-Studio:

V1100. Unreal Engine. Declaring a pointer to a type derived from 'UObject' in a class that is not derived from 'UObject' is dangerous. The pointer may start pointing to an invalid object after garbage collection.

```
public void InsertChar(string c)
{
    . . .
    waiter() ; //avoid double button clicks
}

IEnumerator waiter()
{
    yield return new
        WaitForSecondsRealtime(2) ;
}
```

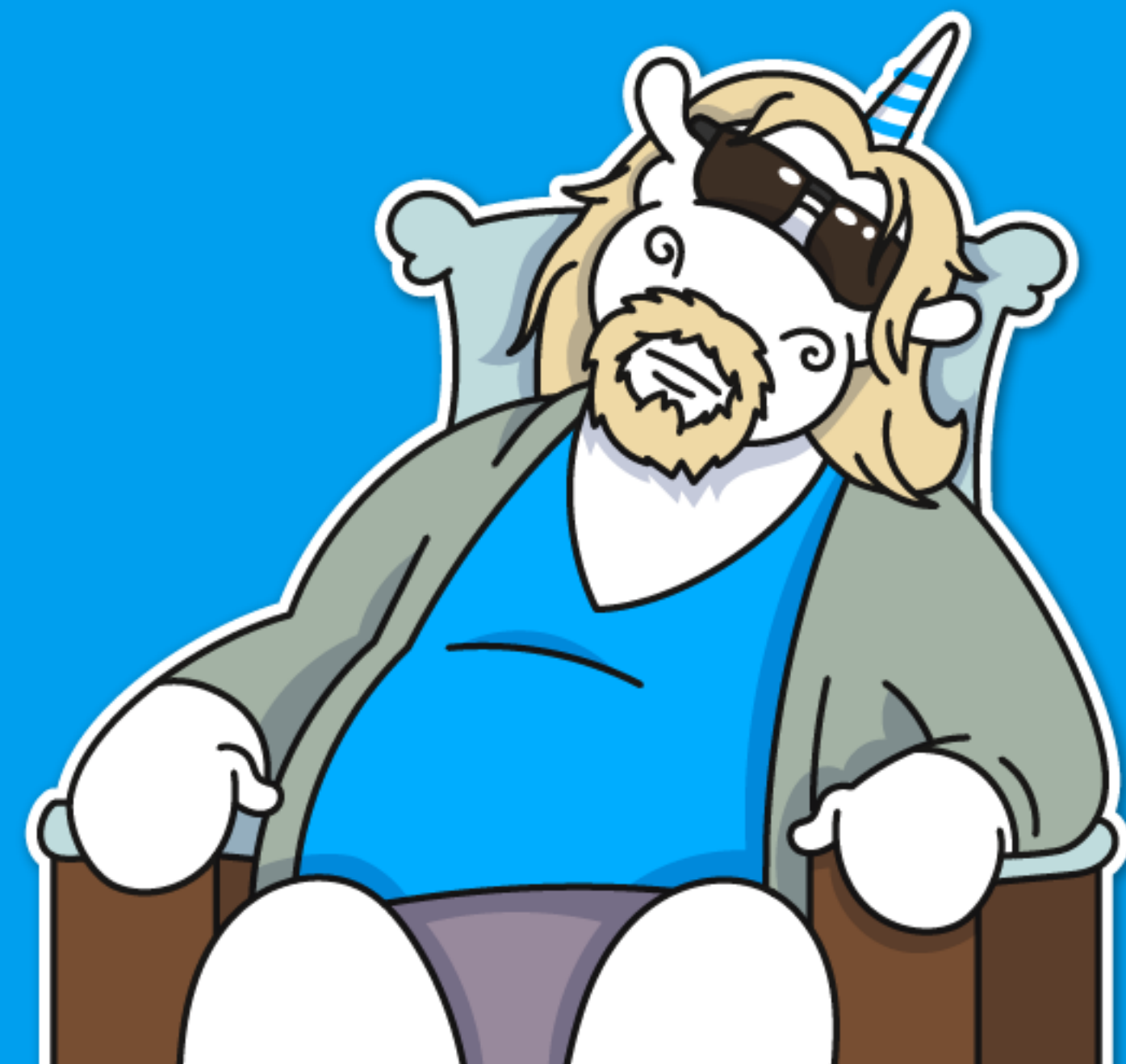
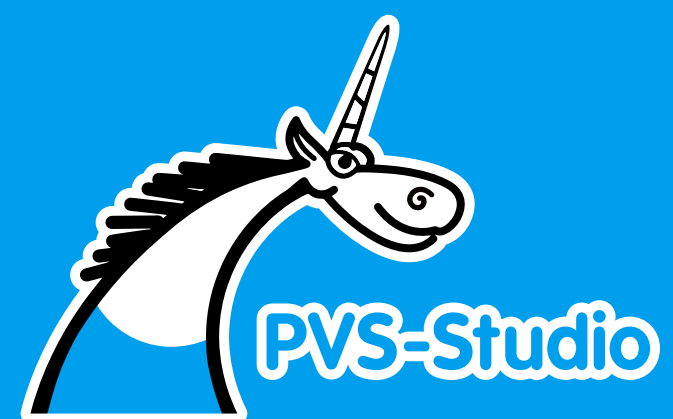
Могут ли статические анализаторы в оптимизацию?



Нет.



Нет.
Почти :)



- Статический анализ под это **не заточен**
- Есть **МИКРОоптимизации**
- Может дополнить профилировщики/
динамические анализаторы



```
inline void setLogTag(const std::string tagName) {  
    m_tag = tagName;  
}
```

```
inline void setLogTag(const std::string tagName) {  
    m_tag = tagName;  
}
```

Предупреждение PVS-Studio:

V801 Decreased performance. It is better to redefine the first function argument as a reference.

Consider replacing 'const .. tagName' with 'const .. &tagName'.


```
inline void setLogTag(const std::string &tagName) {  
    m_tag = tagName;  
}
```

Предупреждение PVS-Studio:

V801 **Decreased performance.** It is better to redefine the first function argument as a reference.

Consider replacing 'const .. tagName' with 'const .. &tagName'.

```
void
addDescriptions (std::vector<std::pair<int, std::string>> toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS) {
        throw std::length_error("Descriptions count would exceed "
                                + std::to_string(MAX_POLICY_DESCRIPTIONS));
    }
    auto addDesc = [] (DescrType **desc, int result,
                      const std::string &name)
    {
        (*desc) = static_cast<DescrType *>(malloc(sizeof(DdescrType)));
        (*desc)->result = result;
        (*desc)->name = strdup(name.data());
    };
    for (const auto &it : toAdd) {
        addDesc(m_policyDescs + m_descCount, it.first, it.second);
        ++m_descCount;
    }
    m_policyDescs[m_descCount] = nullptr;
}
```

```
void
addDescriptions(std::vector<std::pair<int, std::string>> toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS) {
        throw std::length_error("Descriptions count would exceed "
                                + std::to_string(MAX_POLICY_DESCRIPTIONS));
    }
    auto addDesc = [] (DescrType **desc, int result,
                      const std::string &name)
    {
        (*desc) = static_cast<DescrType *>(malloc(sizeof(DdescrType)));
        (*desc)->result = result;
        (*desc)->name = strdup(name.data());
    };
    for (const auto &it : toAdd) {
        addDesc(m_policyDescs + m_descCount, it.first, it.second);
        ++m_descCount;
    }
    m_policyDescs[m_descCount] = nullptr;
}
```

```
void
```

44

```
addDescriptions (std::vector<std::pair<int, std::string>> toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS)
        { . . . . };
    for (const auto &it : toAdd)
        { . . . . };
}
```

```
void
```

45

```
addDescriptions (std::vector<std::pair<int, std::string>> toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS)
        { . . . . };
    for (const auto &it : toAdd)
        { . . . . };
}
```

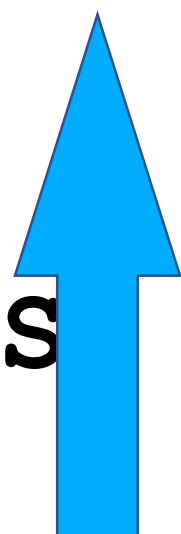


```
addDescriptions (std::vector<std::pair<int, std::string>> toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS)
        { . . . . };
    for (const auto &it : toAdd)
        { . . . . };
}
```

Предупреждение PVS-Studio:

V813 Decreased performance. The 'toAdd' argument should probably be rendered as a constant reference.

```
addDescriptions (std::vector<std::pair<int, std::string>> &toAdd)
{
    if (m_descCount + toAdd.size() > MAX_POLICY_DESCRIPTIONS
        { . . . . };
    for (const auto &it : toAdd)
        { . . . . };
}
```



Предупреждение PVS-Studio:

V813 Decreased performance. The 'toAdd' argument should probably be rendered as a constant reference.

```
private void LateUpdate()
{
    ....
    if (ped != null)
        this.FocusPos = ped.transform.position;

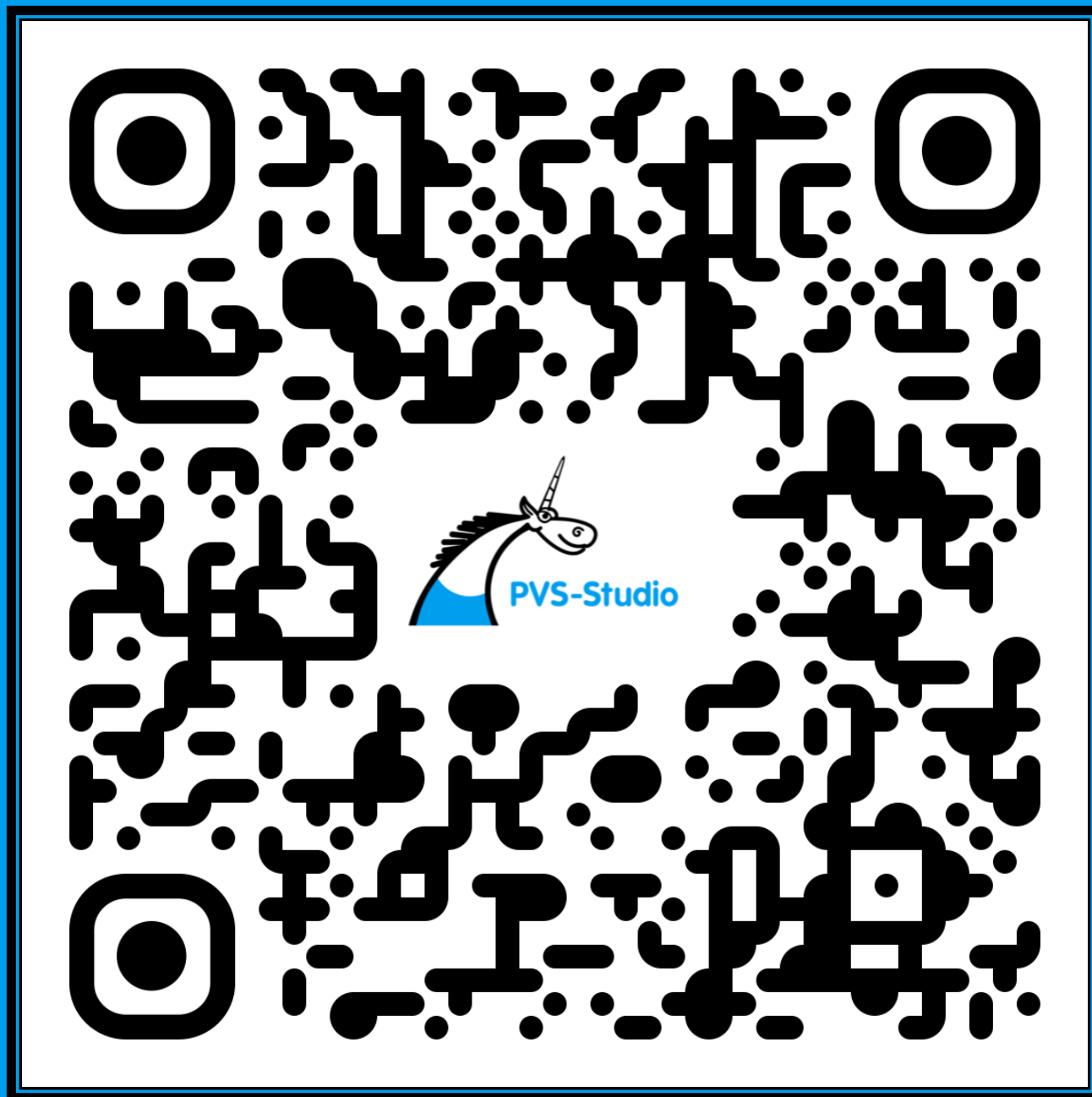
    else if (Camera.main != null)
        this.FocusPos = Camera.main.transform.position;
    ....
    float relAngle = Camera.main != null ?
        Camera.main.transform.eulerAngles.y : 0f;
    ....
}
```

```
private void LateUpdate()  
{  
    ....  
    if (ped != null)  
        this.FocusPos = ped.transform.position;  
  
    else if (Camera.main != null)  
        this.FocusPos = Camera.main.transform.position;  
    ....  
    float relAngle = Camera.main != null ?
```

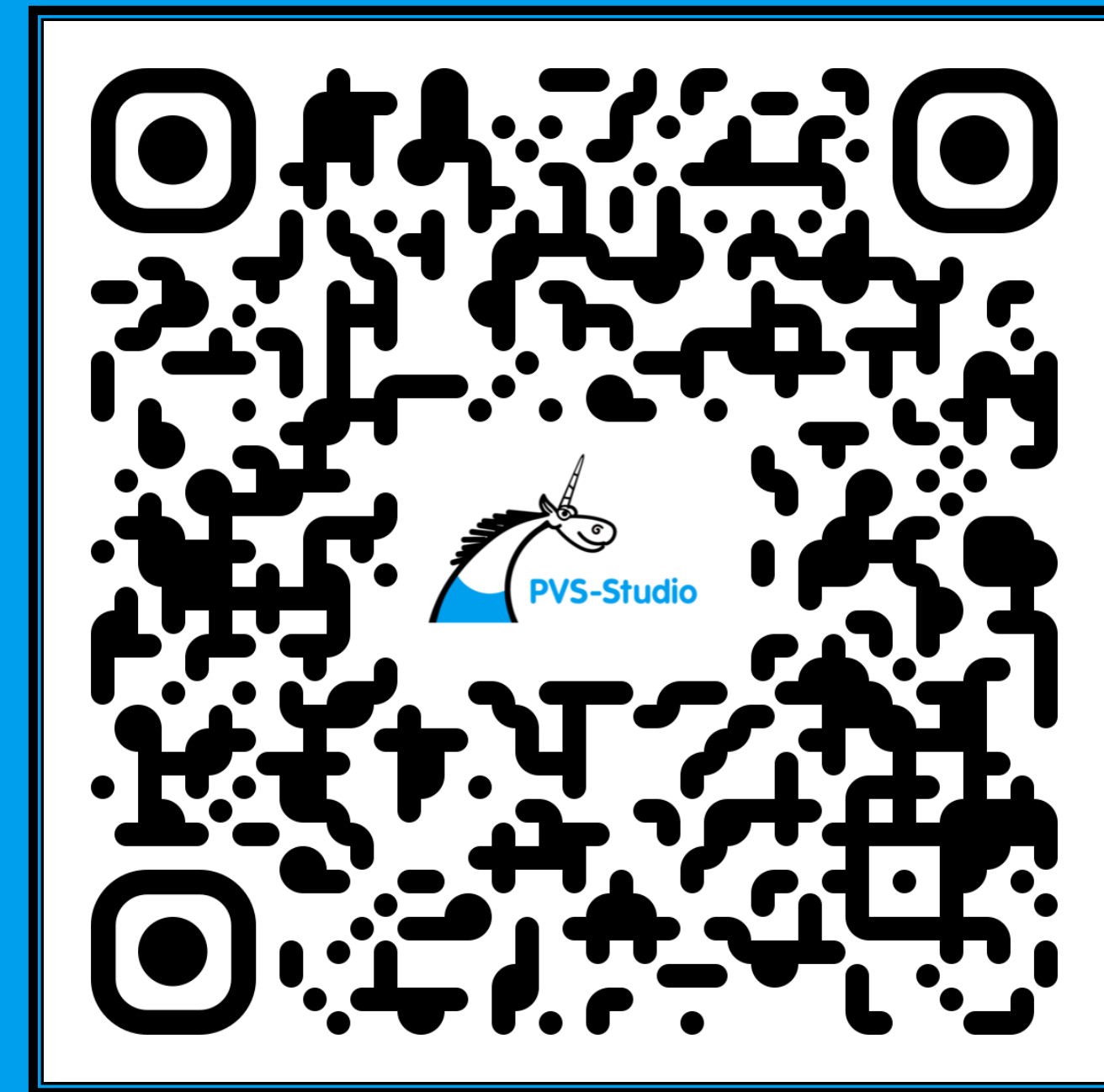
Предупреждение PVS-Studio:

V4005 Expensive operation is performed inside the 'Camera.main' property. Using such property in performance-sensitive context can lead to decreased performance.

Материалы по теме

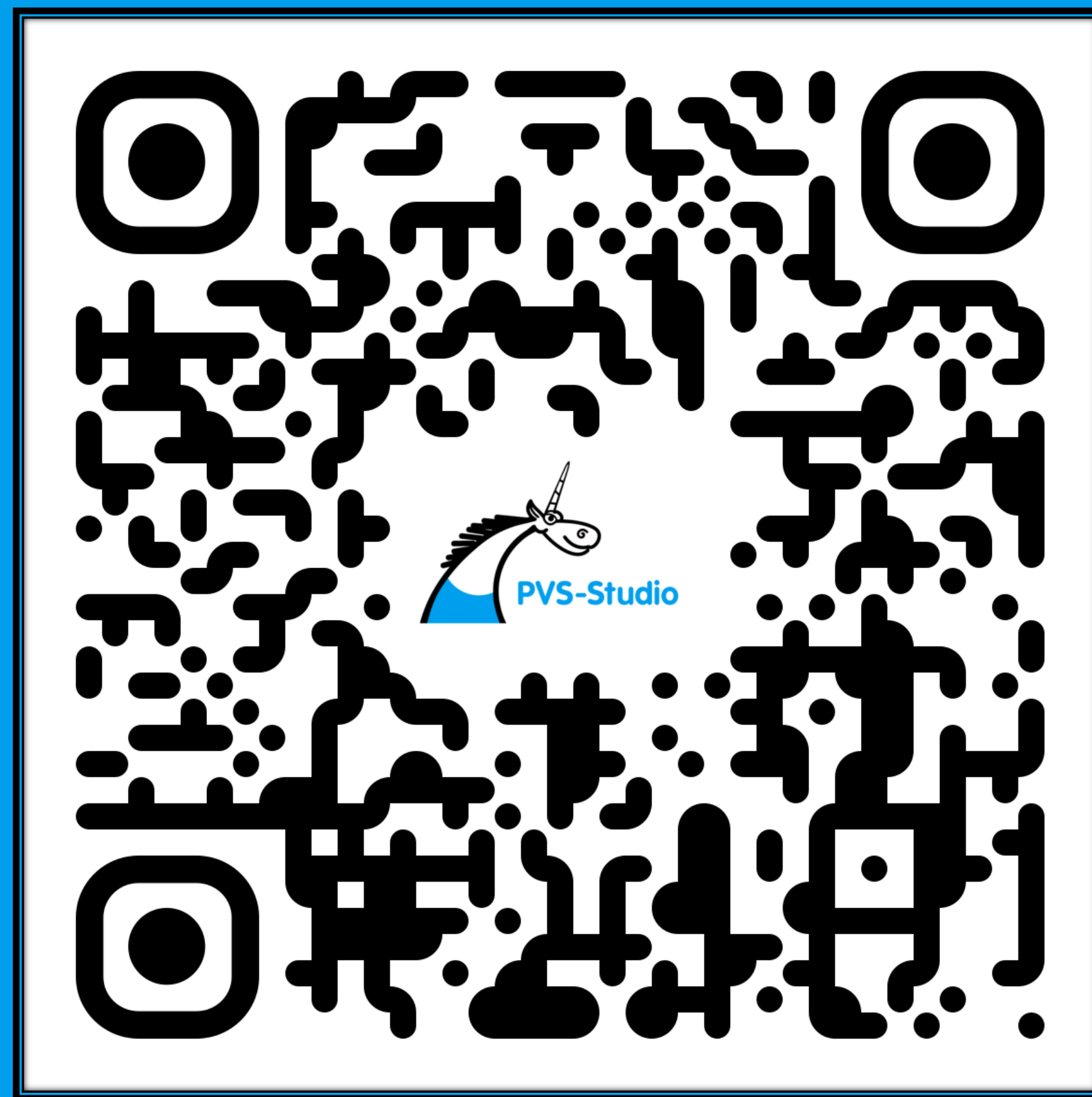


На защите GameDev'а:
статический анализ и Unity



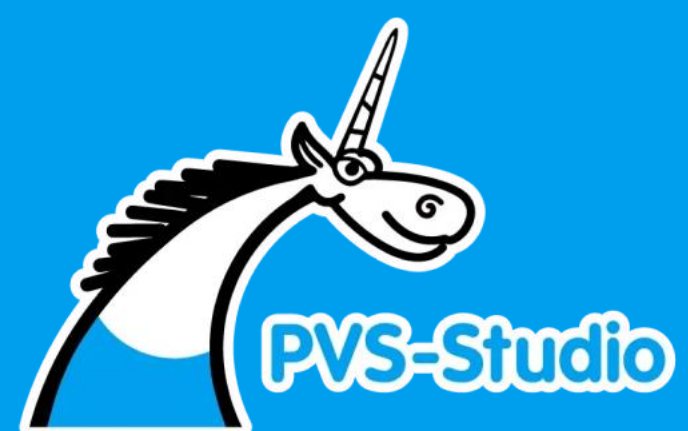
Меньше багов — больше FPS: как
статический анализ помогает
проектам на Unreal Engine

Помогите нам стать лучше!



pvs-studio.ru/ru/about-feedback/

Сделай свой проект
чистым и безопасным
вместе с PVS-Studio



Подпишись на Хабр
Сергея Кушниренко



Стань частью команды
Playrix

