

Как устроено тестирование средства статического тестирования



Сергей Васильев

Тесты

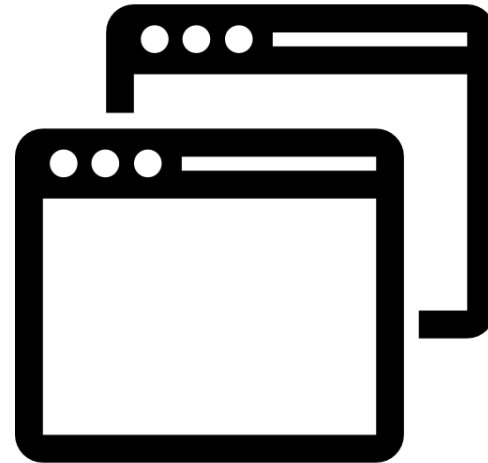
Модульные

Функциональные

Регрессионные

UI

....



Инструменты

SAST

DAST

IAST

SCA

....

Тесты

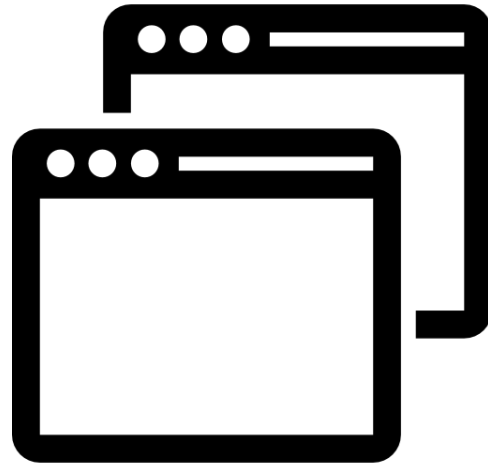
Модульные

Функциональные

Регрессионные

UI

....



Инструменты

SAST

DAST

IAST

SCA

....

Тесты

Модульные

Функциональные

Регрессионные

UI

....

Инструменты

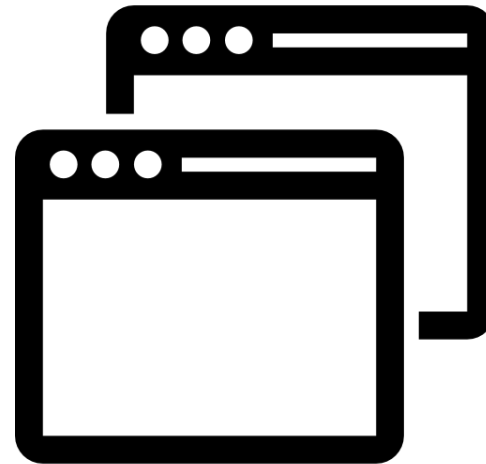
SAST

DAST

IAST

SCA

....



Тесты

Модульные

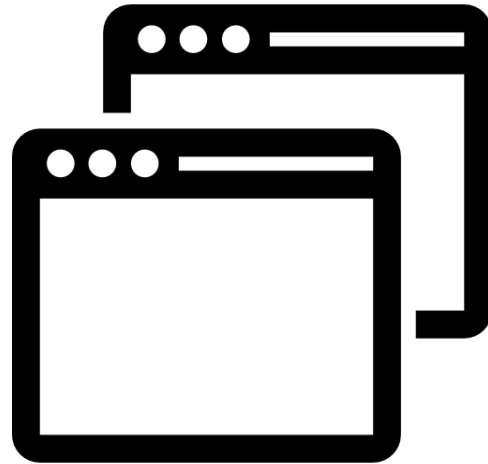
Функциональные

Регрессионные

UI

....

SAST



Инструменты

SAST

DAST

IAST

SCA

....

Тесты

Модульные

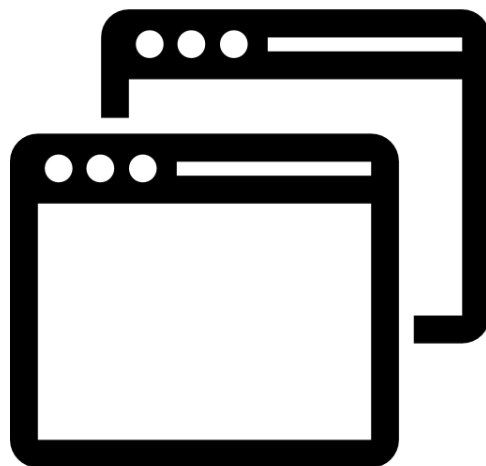
Функциональные

Регрессионные

UI

....

SAST



Инструменты

SAST

DAST

IAST

SCA

....



Статический анализатор vs статический анализатор



Спикер

Сергей Васильев

Head of DevRel в PVS-Studio LLC

7 лет в статическом анализе

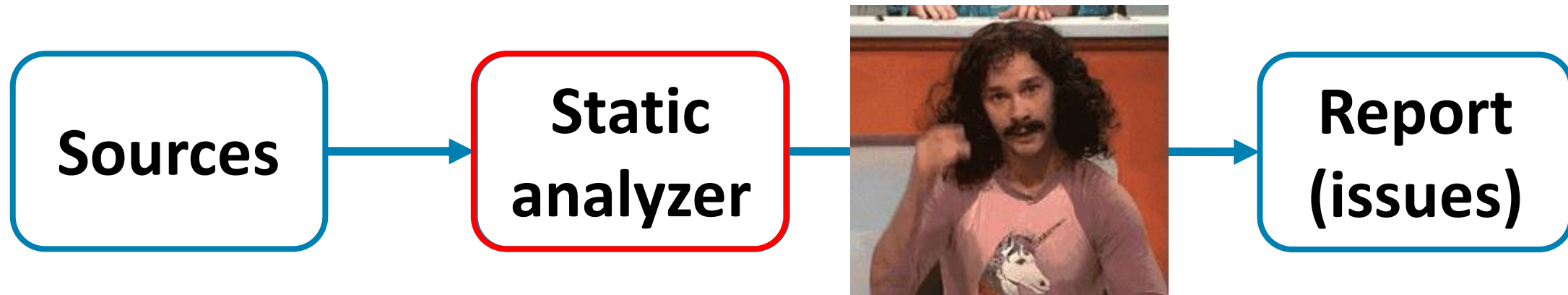
В прошлом:

- C++, C# developer;
- Senior Developer;
- Tools & DevOps Team Leader;
- C# Analyzer Team Leader.

Пишу на [habr](#), выступаю.



Статический анализ



Декомпозиция: продукт

Static analyzer

**Analyzer's
core**

**CLI
analysis tools**

CI/CD plugins

...

IDE plugins

**Issue trackers
plugins**

...

**Build systems
plugins**

Converters

...

Тестирование

Тестируем:

- CLI интерфейсы
- UI плагинов для IDE
- Подавление предупреждений
- Отслеживание запусков компиляторов
- ...

CLI
analysis tools

CI/CD plugins

...

IDE plugins

Issue trackers
plugins

...

Build systems
plugins

Converters

...

Тесты

Модульные

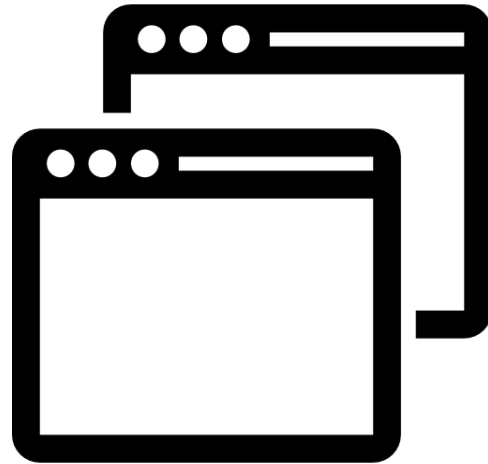
Функциональные

Регрессионные

UI

....

SAST



Инструменты

SAST

DAST

IAST

SCA

....

Тесты

Модульные

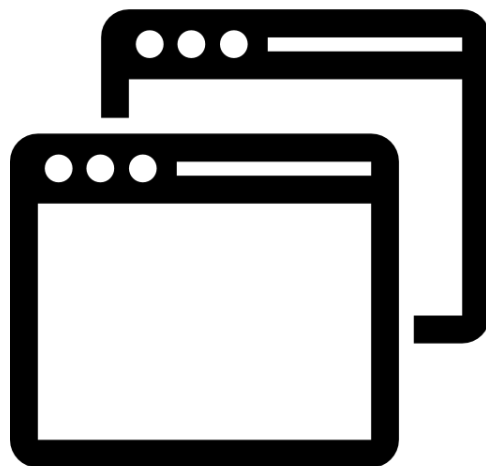
Функциональные

Регрессионные

UI

....

SAST



Инструменты

SAST

DAST

IAST

SCA

....

Декомпозиция: продукт

Static analyzer

**Analyzer's
core**

**CLI
analysis tools**

CI/CD plugins

...

IDE plugins

**Issue trackers
plugins**

...

**Build systems
plugins**

Converters

...

Декомпозиция: ядро анализатора

Analyzer's core

Lexer

Syntax analyzer

Semantic analyzer

Data-flow analysis

**Control-flow
analysis**

Taint analysis

Symbolic execution

**Interprocedural
analysis**

**Intermodular
analysis**

Diagnostic rules

**Function
annotations**

Декомпозиция: ядро анализатора

Analyzer's core

Lexer

Syntax analyzer

Semantic analyzer

Data-flow analysis

Control-flow
analysis

Taint analysis

Symbolic execution

Interprocedural
analysis

Intermodular
analysis

Diagnostic rules

Function
annotations

Декомпозиция: ядро анализатора

Analyzer's core

Lexer

Syntax analyzer

Semantic analyzer

**C#: Roslyn
Java: Spoon**

Data-flow analysis

Control-flow
analysis

Taint analysis

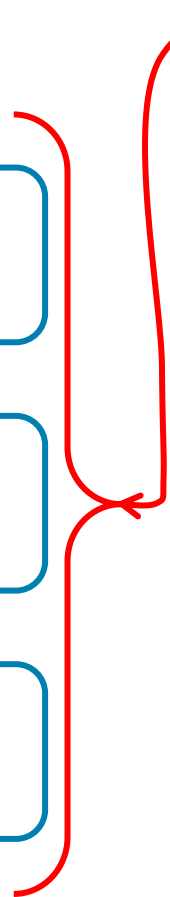
Symbolic execution

Interprocedural
analysis

Intermodular
analysis

Diagnostic rules

Function
annotations



Декомпозиция: ядро анализатора

Analyzer's core

Lexer

Syntax analyzer

Semantic analyzer

Data-flow analysis

**Control-flow
analysis**

Taint analysis

Symbolic execution

**Interprocedural
analysis**

**Intermodular
analysis**

Diagnostic rules

**Function
annotations**

Декомпозиция: ядро анализатора

Analyzer's core

Lexer

Syntax analyzer

Semantic analyzer

Data-flow analysis

Control-flow
analysis

Taint analysis

Symbolic execution

Interprocedural
analysis

Intermodular
analysis

Diagnostic rules

Function
annotations

Типы предупреждений

Какие бывают предупреждения?

		Ругается	Не ругается
		Positive	Negative
Ожидаемо	True	True positive	True negative
Не ожидаемо	False	False positive	False negative

Какие бывают предупреждения?

		Ругается	Не ругается
		Positive	Negative
Ожидаемо	True	True positive	True negative
Не ожидаемо	False	False positive	False negative

Какие бывают предупреждения?

		Ругается	Не ругается
		Positive	Negative
Ожидаемо	True	True positive	True negative
Не ожидаемо	False	False positive	False negative

Какие бывают предупреждения?

		Ругается	Не ругается
		Positive	Negative
Ожидаемо	True	True positive	True negative
Не ожидаемо	False	False positive	False negative

Какие бывают предупреждения?

		Ругается	Не ругается
		Positive	Negative
Ожидаемо	True	True positive	True negative
Не ожидаемо	False	False positive	False negative

Same then/else branches of the 'if' statement

```
if (condition)
{
    Bad();
}
else
{
    Bad();
}
```

```
if (condition)
{
    Good();
}
else
{
    Bad();
}
```

Типы предупреждений

```
if (condition)
{
    Bad();
}
else
{
    Bad();
}
```

Issue

V3004 The 'then' statement is equivalent to the 'else' statement.

Ожидается?	
Ругается?	

**True
positive**

Типы предупреждений

```
if (condition)
{
    Bad();
}
else
{
    Bad();
}
```

Ожидается?	✗
Ругается?	✗

**False
negative**

Типы предупреждений

```
if (condition)
{
    Good();
}
else
{
    Bad();
}
```

Ожидается?	✓
Ругается?	✗

**True
negative**

Типы предупреждений

```
if (condition)
{
    Good();
}
else
{
    Bad();
}
```

Issue

V3004 The 'then' statement is equivalent to the 'else' statement.

Ожидается?	
Ругается?	

**False
positive**

Что делать?

Увеличиваем **положительные** срабатывания:

- true positives
- true negative

Уменьшаем **негативные** срабатывания:

- false positives
- false negatives



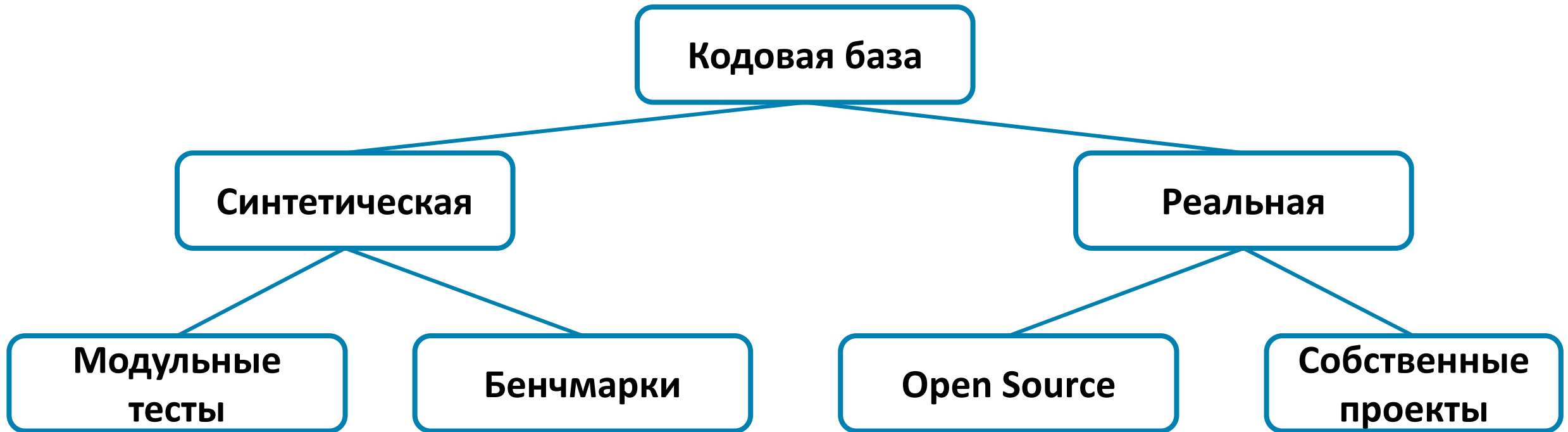
Где нужно – ругаемся,
где не нужно – не ругаемся

Где нужно – ругаемся,
где не нужно – не ругаемся

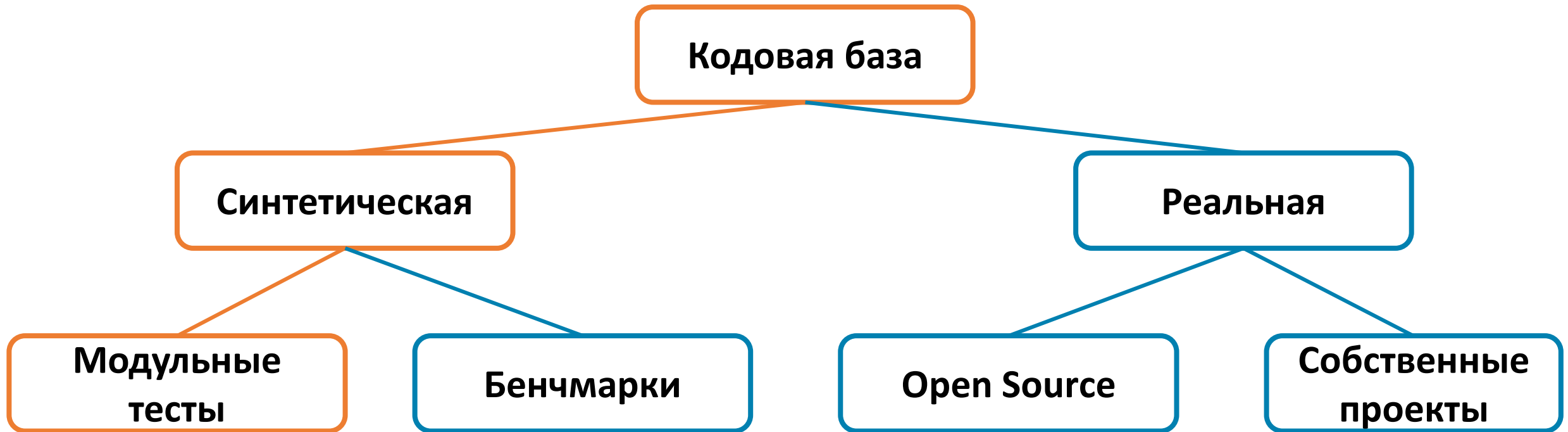
Как?

Тесты на коде

На чём тестировать?



На чём тестировать?

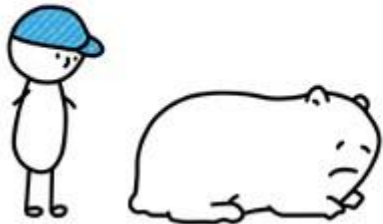


Модульные тесты

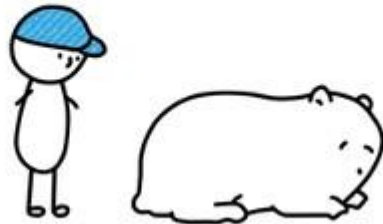
Модульные тесты

- TDD в действии
 - Составляем наборы:
 - true positives
 - true negatives
- Не проходит? *False negative*
Не проходит? *False positive*

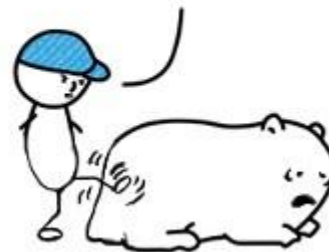
Разработчик



Тесты



Проходите давайте



A == A

```
if (A == A)
{ }
```

```
if (A == (A))
{ }
```

```
if (this.A == this.A)
{ }
```

```
if (A == this.A)
{ }
```

```
if (this.A == base.A)
{ }
```

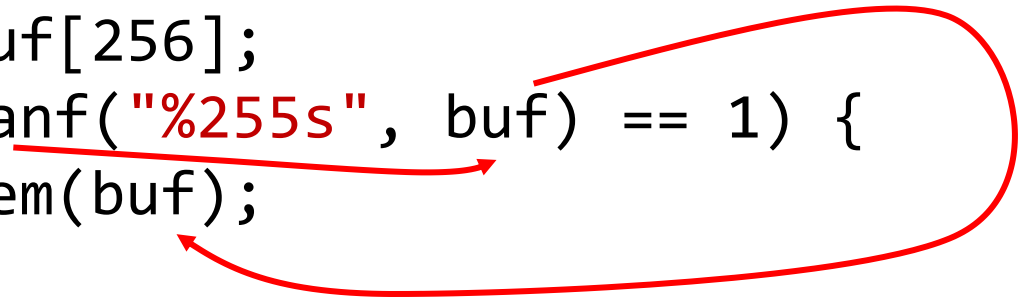
```
if (this.A == (base.A))
{ }
```

```
if (((this.A)) == (base.A))
{ }
```

```
if (((this.A)) != (base.A))
{ }
```

Taint analysis

```
static void F1()  
{  
    char buf[256];  
    if (scanf("%255s", buf) == 1) {  
        system(buf);  
    }  
}
```



Taint analysis

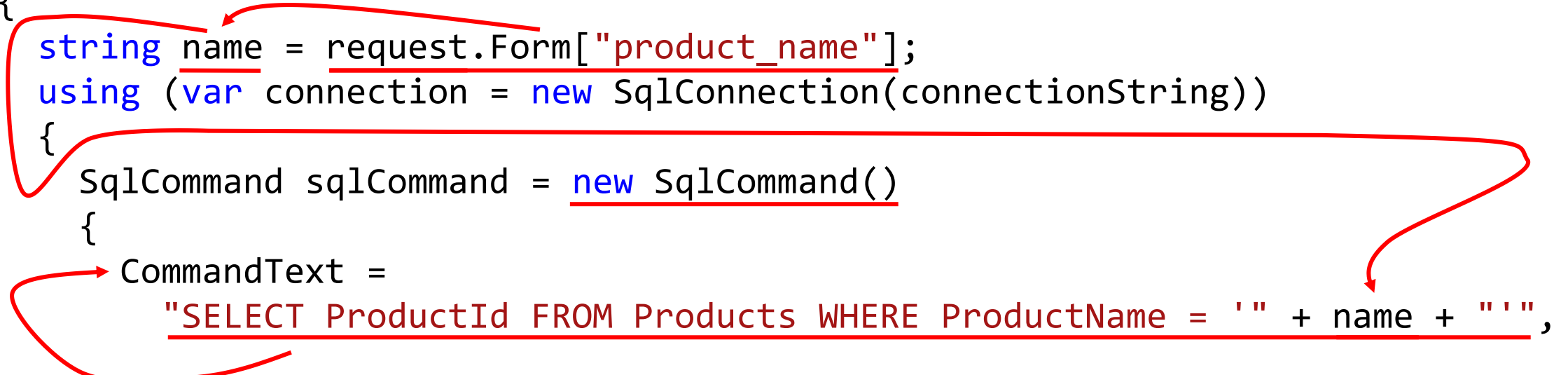
```
char readbuf[BUFSIZ];  
if (fgets(readbuf, BUFSIZ, stdin) == NULL)  
{  
    if (feof(stdin))  
        return 0;  
  
    return -1;  
}  
  
if (readbuf[strlen(readbuf) - 1] == '\n')  
    readbuf[strlen(readbuf) - 1] = '\0';
```

The diagram illustrates a data flow in a C program. A red arrow originates from the `stdin` argument in the `fgets` function call and points to the `strlen(readbuf)` expression within the second `if` statement. This indicates that the length of the buffer, which is determined by the input from `stdin`, is used to access the last element of the `readbuf` array.

SQLI

```
void InitializerTest(HttpRequest request, String connectionString)
{
    string name = request.Form["product_name"];
    using (var connection = new SqlConnection(connectionString))
    {
        SqlCommand sqlCommand = new SqlCommand()
        {
            CommandText =
                "SELECT ProductId FROM Products WHERE ProductName = '" + name + "'",
            CommandType = CommandType.Text
        };

        SqlDataReader reader = sqlCommand.ExecuteReader();
    }
}
```



SQLI

```
void InitializerTest(HttpRequest request, String connectionString)
{
    string name = request.Form["product_name"];
    using (var connection = new SqlConnection(connectionString))
    {
        SqlCommand sqlCommand = new SqlCommand()
        {
            CommandText =
                "SELECT ProductId FROM Products WHERE ProductName = '" + name + "'",

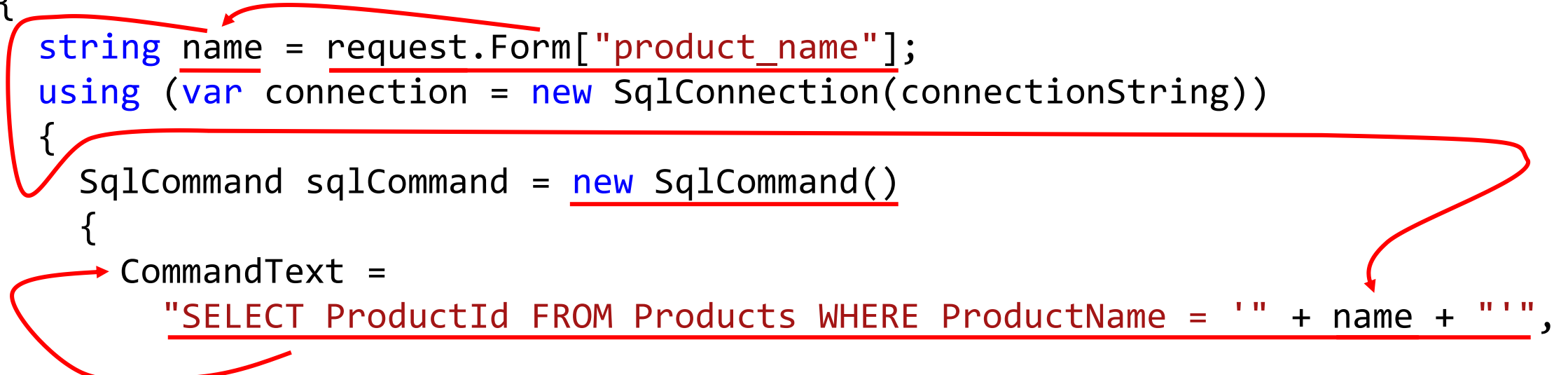
            CommandType = CommandType.Text
        };

        SqlDataReader reader = sqlCommand.ExecuteReader();
    }
}
```

SQLI

```
void InitializerTest(HttpRequest request, String connectionString)
{
    string name = request.Form["product_name"];
    using (var connection = new SqlConnection(connectionString))
    {
        SqlCommand sqlCommand = new SqlCommand()
        {
            CommandText =
                "SELECT ProductId FROM Products WHERE ProductName = '" + name + "'",
            CommandType = CommandType.Text
        };

        SqlDataReader reader = sqlCommand.ExecuteReader();
    }
}
```



XXE

```
private static string
ParseRequest(System.Web.HttpContext context)
{
    var buffer = new byte[context.Request
                           .InputStream.Length];
    context.Request.InputStream.Read(buffer,
                                     0,
                                     buffer.Length);

    return Encoding.Default.GetString(buffer);
}

void CVE_2018_14485(System.Web.HttpContext input)
{
    var inputXml = ParseRequest(input);

    LoadXmlRequest(inputXml); //+V5614
}
```

```
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument()
    {
        XmlResolver = new XmlUrlResolver()
    };

    try
    {
        if (!(xml.StartsWith("<?xml") ||
                    xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }

        request.LoadXml(xml);
    }
    catch (Exception ex)
    {
        throw new Exception(/*...*/);
    }
    //....
}
```

XXE: CVE-2018-14485

```
private static string
ParseRequest(System.Web.HttpContext context)
{
    var buffer = new byte[context.Request
                          .InputStream.Length];
    context.Request.InputStream.Read(buffer,
                                     0,
                                     buffer.Length);

    return Encoding.Default.GetString(buffer);
}

void CVE_2018_14485(System.Web.HttpContext input)
{
    var inputXml = ParseRequest(input);
    LoadXmlRequest(inputXml); //+V5614
}
```

The diagram illustrates the flow of data in the CVE-2018-14485 exploit. Red arrows show the path from the `CVE_2018_14485` method to `ParseRequest`, and then from `ParseRequest` to `LoadXmlRequest`. A red box highlights `context.Request.InputStream` in `ParseRequest`, and another red box highlights `request.LoadXml(xml)` in `LoadXmlRequest`. The `LoadXmlRequest` method is also highlighted with a large red oval.

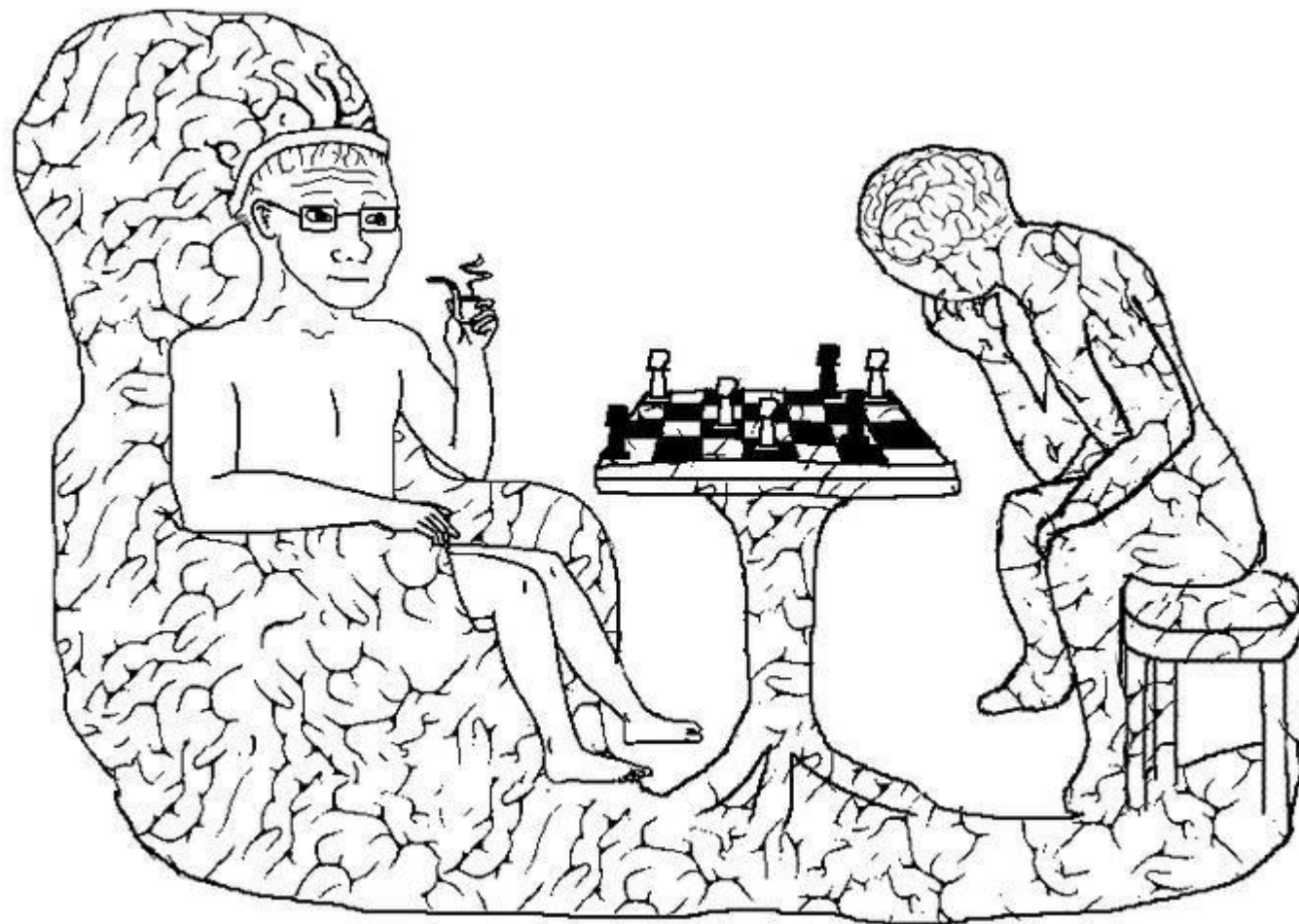
```
private void LoadXmlRequest(string xml)
{
    var request = new XmlDocument()
    {
        XmlResolver = new XmlUrlResolver()
    };

    try
    {
        if (!(xml.StartsWith("<?xml") ||
                        xml.StartsWith("<method")))
        {
            xml = xml.Substring(xml.IndexOf("<?xml"));
        }

        request.LoadXml(xml);
    }
    catch (Exception ex)
    {
        throw new Exception(/*...*/);
    }
    //....
}
```

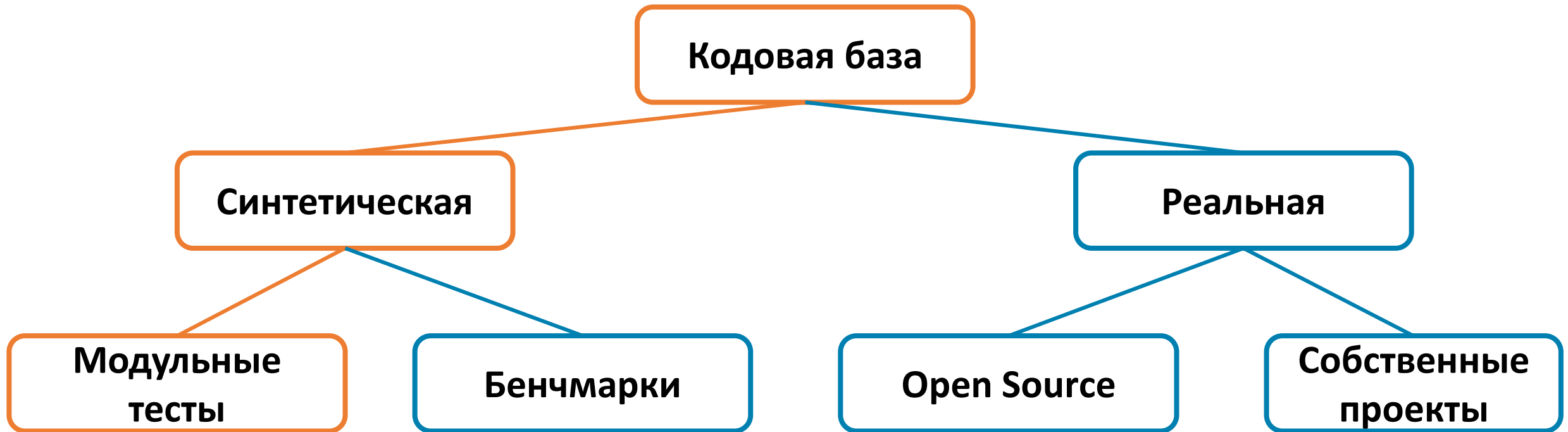
The diagram illustrates the flow of data in the `LoadXmlRequest` method. Red arrows show the path from the `xml` parameter to `request.LoadXml(xml)`. A red box highlights `request.LoadXml(xml)`. The `LoadXmlRequest` method is also highlighted with a large red oval.

Когда придумываешь синтетические примеры уязвимостей

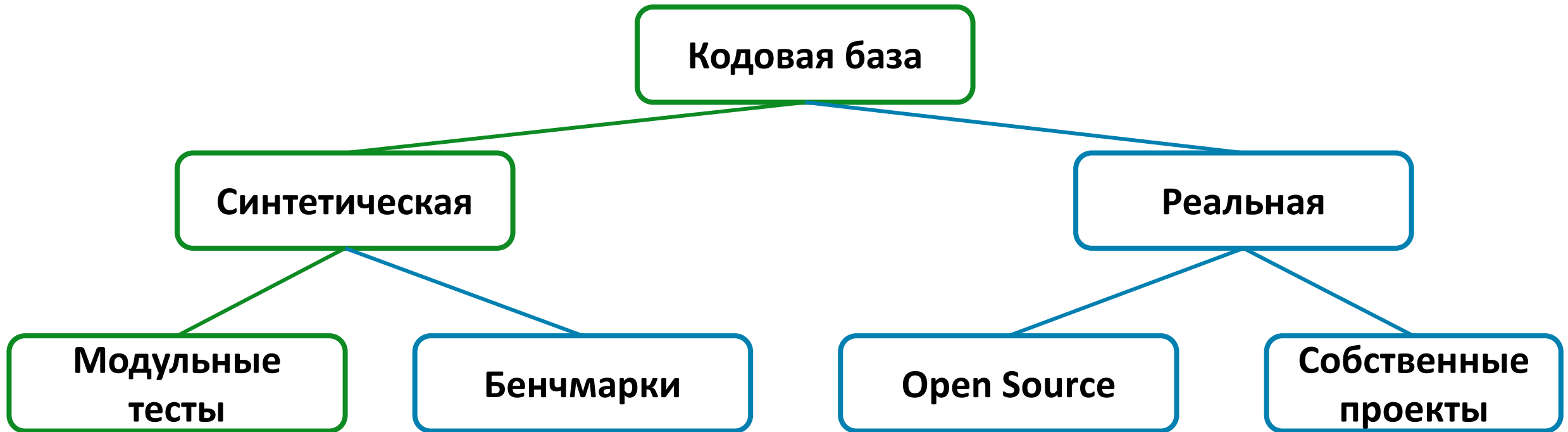


Синтетика подойдёт для старта.
Но её недостаточно.

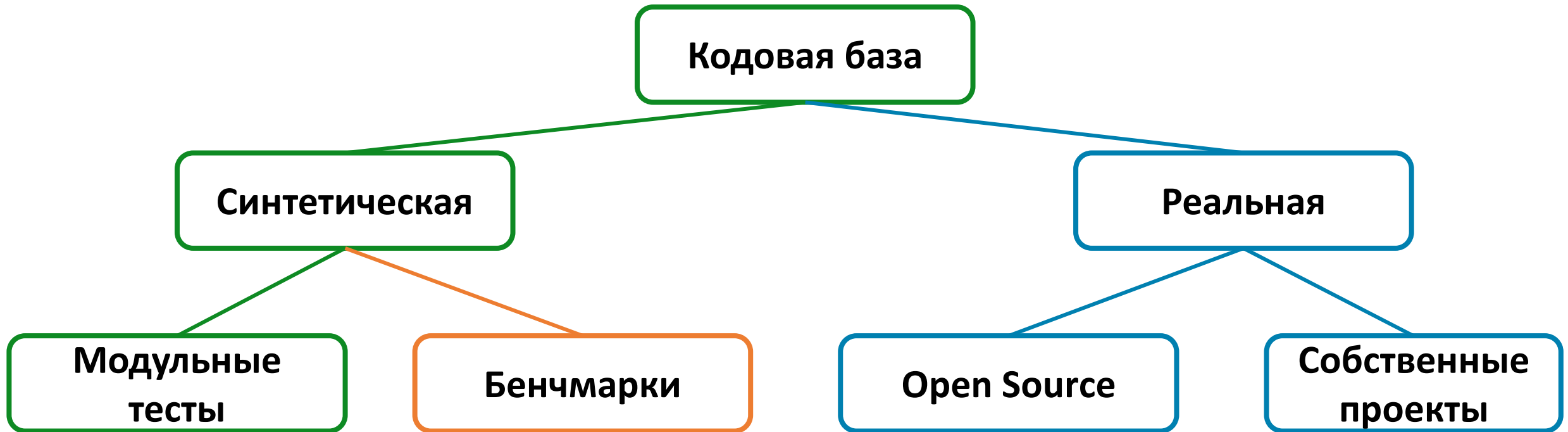
На чём тестировать?



На чём тестировать?



На чём тестировать?



Бенчмарки стат. анализаторов

Бенчмарки стат. анализаторов

Бенчмарк	
Позитивные тесты	Негативные тесты

- Оценка покрытия
- “Тренировка анализатора”
- Сравнение инструментов

Примеры:

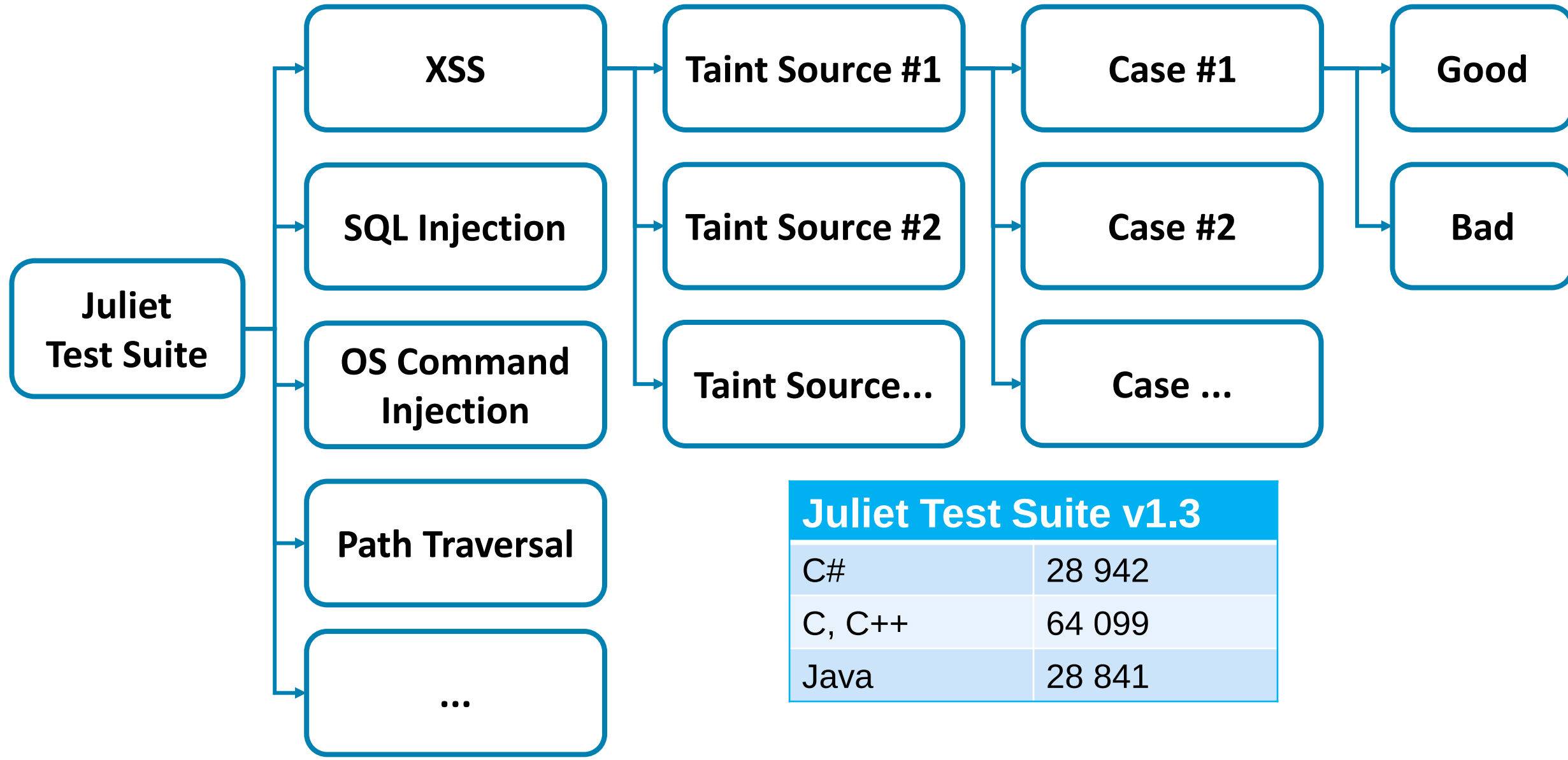
- Juliet Test Suite;
- Toyota ITC Benchmark;
- PHP Vulnerability Test Suite;

Juliet Test Suite

- Test suites от NIST (National Institute of Standards and Technology)
- Тест-кейсы для разных CWE:
 - C#: 105
 - C/C++: 118
 - Java: 112



Juliet Test Suite



Juliet Test Suite v1.3

C#	28 942
C, C++	64 099
Java	28 841

Juliet Test Suite: CWE-89 SQLI - Bad

```
public override void Bad(HttpRequest req, HttpResponse resp) {
    string data;
    data = ""; /* Initialize data */
    /* Read data using an outbound tcp connection */
    {
        try {
            /* Read data using an outbound tcp connection */
            using (TcpClient tcpConn = new TcpClient("host.example.org", 39544)) {
                /* read input from socket */
                using (StreamReader sr = new StreamReader(tcpConn.GetStream())) {
                    /* POTENTIAL FLAW: Read data using an outbound tcp connection */
                    data = sr.ReadLine();
                }
            }
        }
        catch (IOException exceptIO) {
            IO.Logger.Log(NLog.LogLevel.Warn, exceptIO, "Error with stream reading");
        }
    }
    if (data != null) {
        string[] names = data.Split('-');
        int successCount = 0;
        SqlCommand badSqlCommand = null;
        try {
            using (SqlConnection dbConnection = IO.GetDBConnection())
            {
                badSqlCommand.Connection = dbConnection;
                dbConnection.Open();
                for (int i = 0; i < names.Length; i++) {
                    /* POTENTIAL FLAW: data concatenated into SQL statement used in CommandText, which could result in SQL Injection */
                    badSqlCommand.CommandText += "update users set hitcount=hitcount+1 where name='" + names[i] + "'";
                }
                var affectedRows = badSqlCommand.ExecuteNonQuery();
                successCount += affectedRows;
                IO.WriteLine("Succeeded in " + successCount + " out of " + names.Length + " queries.");
            }
        }
        catch (SqlException exceptSql) {
            IO.Logger.Log(NLog.LogLevel.Warn, "Error getting database connection", exceptSql);
        }
        finally {
            try {
                if (badSqlCommand != null) {
                    badSqlCommand.Dispose();
                }
            }
            catch (SqlException exceptSql) {
                IO.Logger.Log(NLog.LogLevel.Warn, "Error disposing SqlCommand", exceptSql);
            }
        }
    }
}
```

Juliet Test Suite: CWE-89 SQLI - *Bad*

```
public override void Bad(HttpRequest req, HttpResponse resp)
{
    data = ReadUsingTcpConnection(); // Taint source
    names = data.Split('-');

    SqlCommand badCommand;

    for (int i = 0; i < names.Length; ++i)
    {
        /* POTENTIAL FLAW: data concatenated into SQL statement used in CommandText,
           which could result in SQL Injection */
        badCommand.CommandText
            += "update users set hitcount=hitcount+1 where name='" + names[i] + "'";
    }

    rows = badCommand.Execute()
}
```

Juliet Test Suite: CWE-89 SQLI - *Bad*

```
public override void Bad(HttpRequest req, HttpResponse resp)
{
    data = ReadUsingTcpConnection(); // Taint source
    names = data.Split('-');

    SqlCommand badCommand;

    for (int i = 0; i < names.Length; ++i)
    {
        /* POTENTIAL FLAW: data concatenated into SQL statement used in CommandText,
           which could result in SQL Injection */
        badCommand.CommandText
            += "update users set hitcount=hitcount+1 where name='" + names[i] + "';";
    }

    rows = badCommand.Execute()
}
```

The diagram illustrates the flow of tainted data from the `ReadUsingTcpConnection()` method to the `names` array, which is then used in the `SqlCommand.CommandText` property. A red box highlights the `ReadUsingTcpConnection()` call, and another red box highlights the string concatenation in the `+=` statement. Red arrows show the data flow from the first box to the `names` variable and then to the string concatenation. A large red arrow points from the `names` variable to the `+=` statement, emphasizing the potential for SQL injection.

Juliet Test Suite: CWE-89 SQLI - Good

```
public override void Good(HttpRequest req, HttpResponse resp)
{
    data = "data";
    names = data.Split('-');

    SqlCommand badCommand;

    for (int i = 0; i < names.Length; ++i)
    {

        badCommand.CommandText
            += "update users set hitcount=hitcount+1 where name='" + names[i] + "';";
    }

    rows = badCommand.Execute()
}
```

Juliet Test Suite: CWE-89 SQLI - Good

```
public override void Good(HttpRequest req, HttpResponse resp)
{
    data = ReadUsingTcpConnection(); // Taint source
    names = data.Split('-');

    SqlCommand goodSqlCommand;

    for (int i = 0; i < names.Length; ++i)
    {
        SqlParameter nameParam = new SqlParameter("@name", SqlDbType.VarChar, 100);
        nameParam.Value = names[i];
        goodSqlCommand.CommandText
            += "update users set hitcount=hitcount+1 where name=@name;";
    }

    rows = badCommand.Execute()
}
```

The diagram illustrates the flow of tainted data from the `ReadUsingTcpConnection()` method to the SQL query string. A red box highlights the `ReadUsingTcpConnection()` call. Red arrows show the data being assigned to `data`, then split into `names`, and finally used in the `SqlCommand` object's `CommandText` property within the `for` loop. A large red oval encircles the `SqlCommand` object and the `for` loop, indicating the overall context of the SQL query construction.

Toyota ITC Benchmarks

Бенчмарк от Toyota InfoTechnology Center для C/C++

Прим. С сайта NIST SAMATE:

*Please note that test cases contain coincidental weaknesses flagged by SAMATE team, each described accordingly and **individually**.*

*Also please note that the SAMATE team determined that in a few cases, the code that was marked as weakness originally was in fact correct code. We describe these cases accordingly and **individually**.*



Toyota ITC Benchmarks: dead code

```
void dead_code_002 ()
{
    int flag = 0;
    int a = 0;
    int ret;
    if (flag)
    {
        a ++; /*Tool should detect this line as error*/
               /*ERROR:Dead Code*/
    }
    ret = a;
    sink = ret;
}
```

Toyota ITC Benchmarks: dead code

```
void dead_code_002 ()
{
    int flag = 0;
    int a = 0;
    int ret;
    if (flag)
    {
        a ++; /*Tool should detect this line as error*/
               /*ERROR:Dead Code*/
    }
    ret = a;
    sink = ret;
}
```

Toyota ITC Benchmarks: dead code

```
void dead_code_002 ()
{
    int flag = 0;
    int a = 0;
    int ret;
    → if (flag) // V547: expression is always false
    {
        → a++; /*Tool should detect this line as error*/
           /*ERROR:Dead Code*/
    }
    ret = a;
    sink = ret;
}
```

Toyota ITC Benchmarks: dead code

```
void dead_code_001 ()
{
    int a = 0;
    int ret;
    if (0) // No warn V547: expression is always false
    {
        a ++; /*Tool should detect this line as error*/
               /*ERROR:Dead Code*/
    }
    ret = a;
    sink = ret;
}
```

Бенчмарки: выводы

Полезны, но без фанатизма

Бенчмарки: выводы



Анализатор
на бенчмарках



Анализатор
на реальных проектах

Бенчмарки: выводы

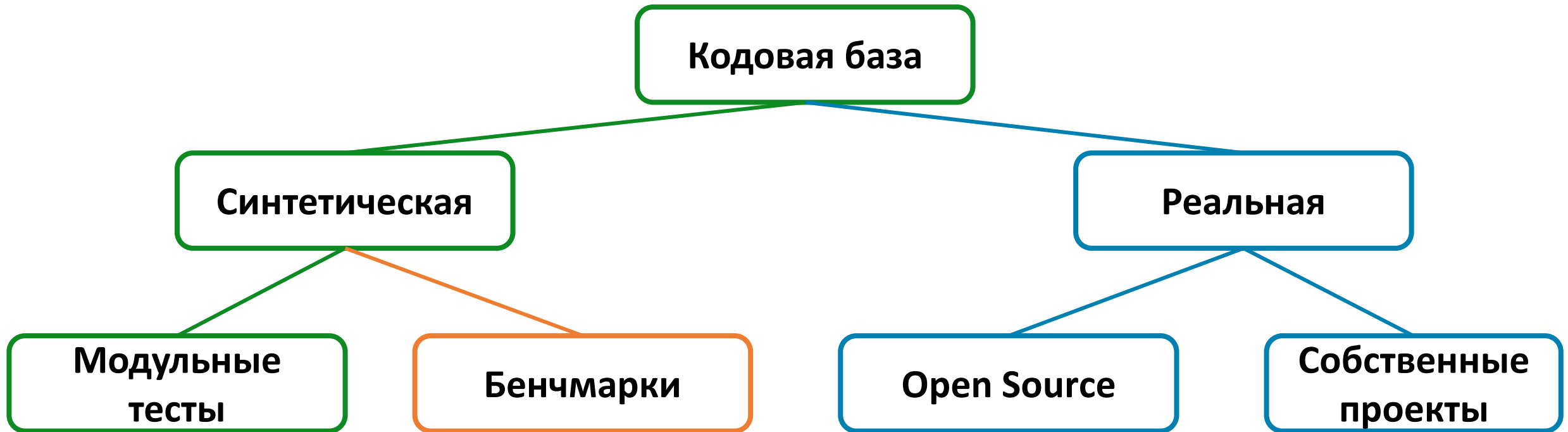


Анализатор
на бенчмарках

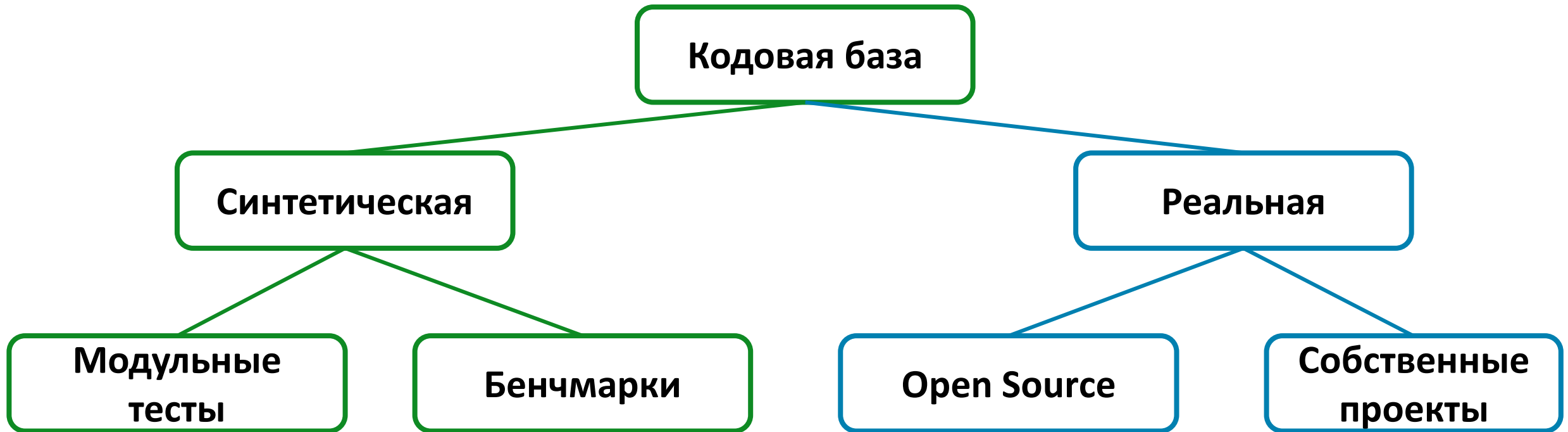


Анализатор
на реальных проектах

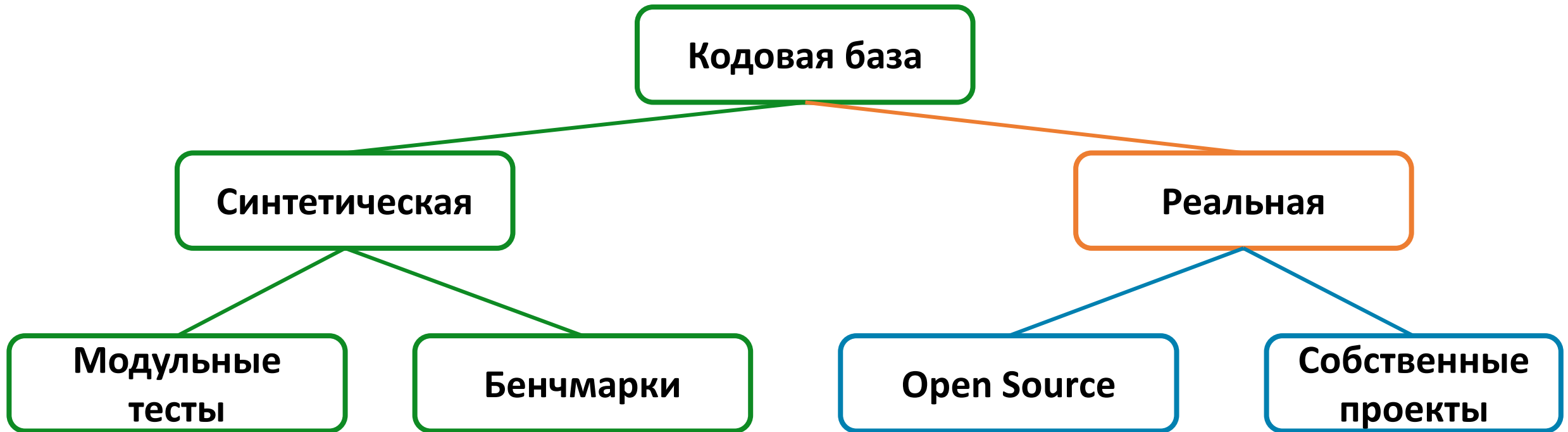
На чём тестировать?



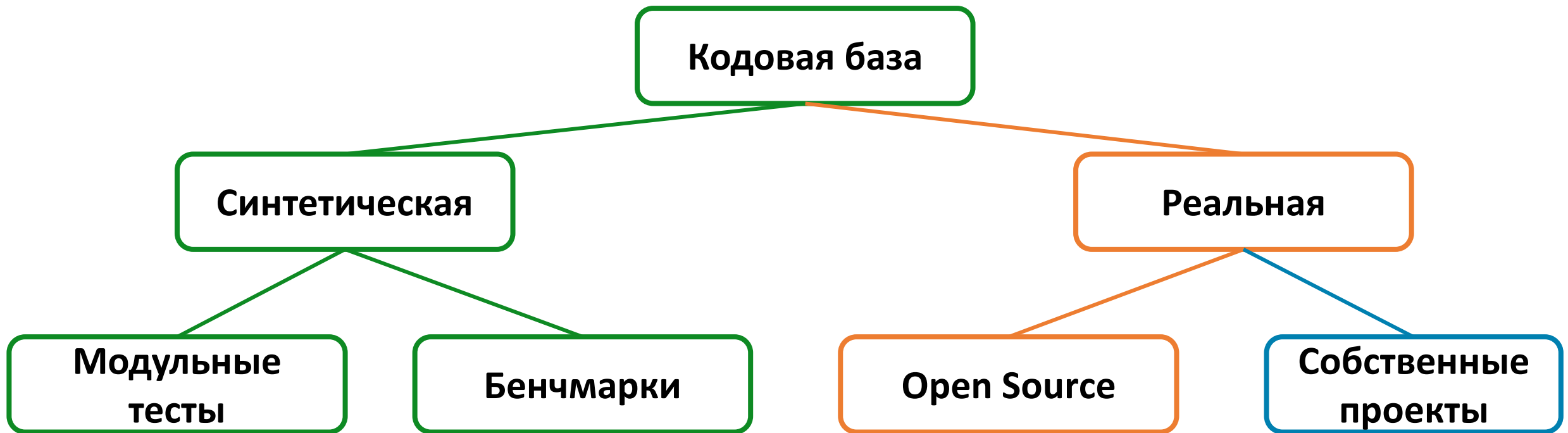
На чём тестировать?



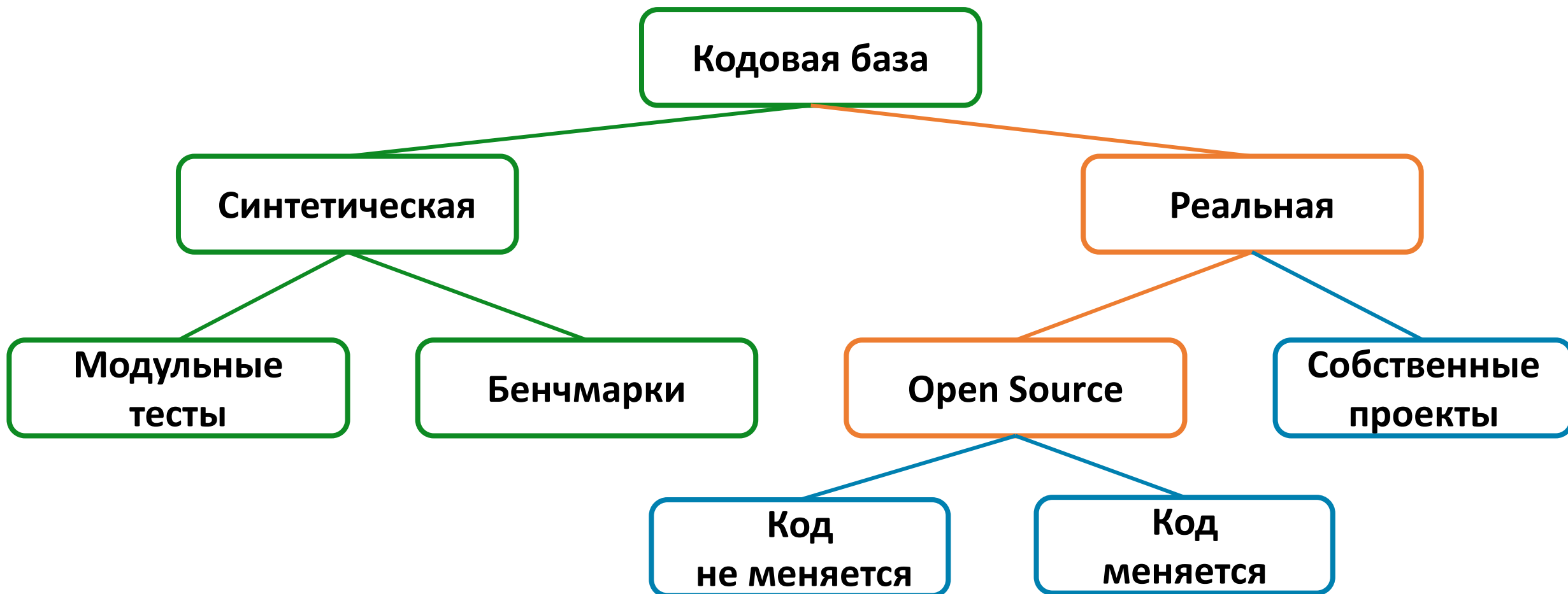
На чём тестировать?



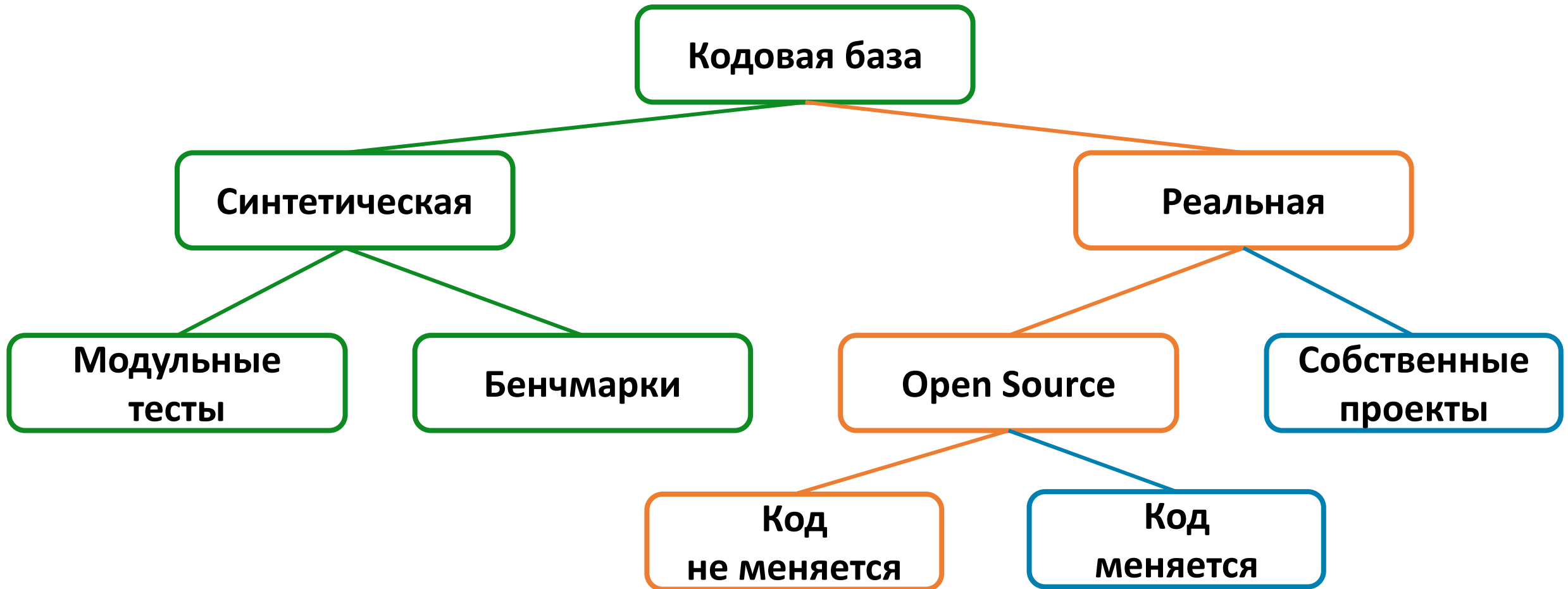
На чём тестировать?



На чём тестировать?



На чём тестировать?



Тестирование на Open Source проектах

Тестирование на OSS: зачем?

```
....  
while (    service is IServiceHolder<TService>  
        && caller is TService callerService)  
{  
    return callerService;  
}  
....
```

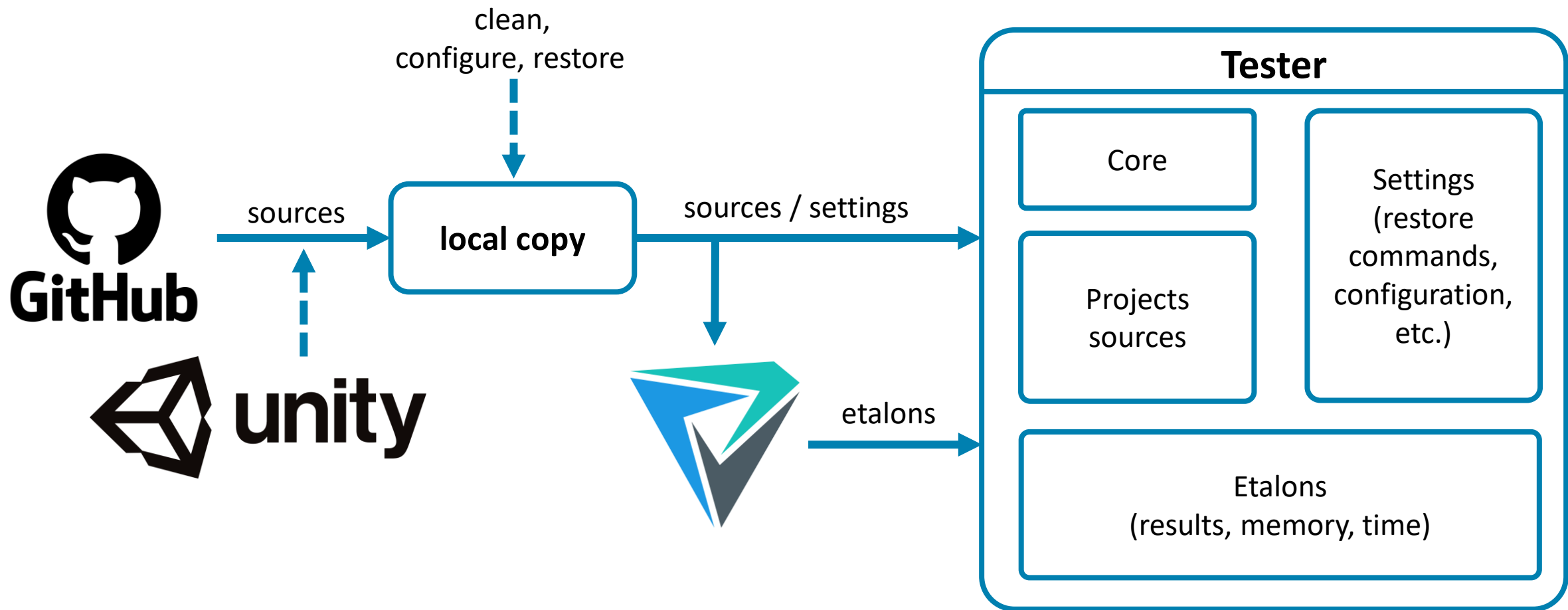


Тестирование на OSS: зачем?

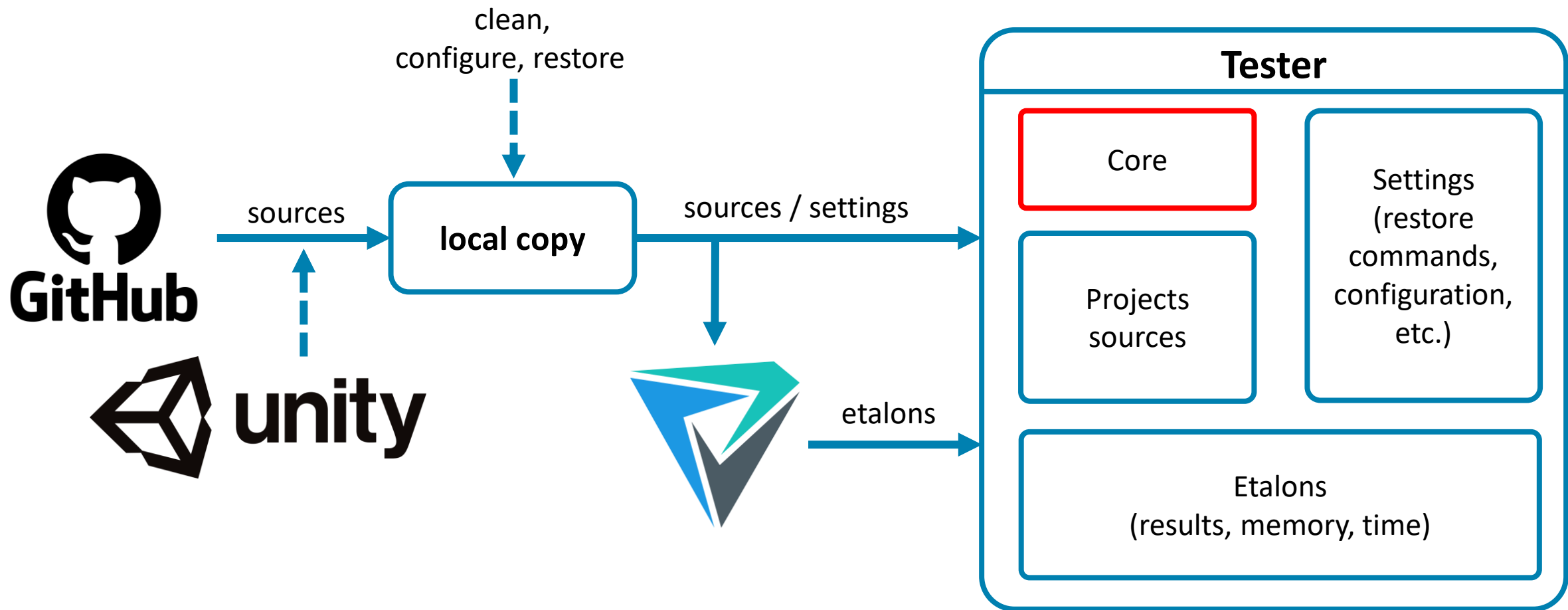
- Реальные кейсы использования:
 - разных фич языка
 - подходов
 - фреймворков
- Тестирование анализатора на условиях, близких к реальным



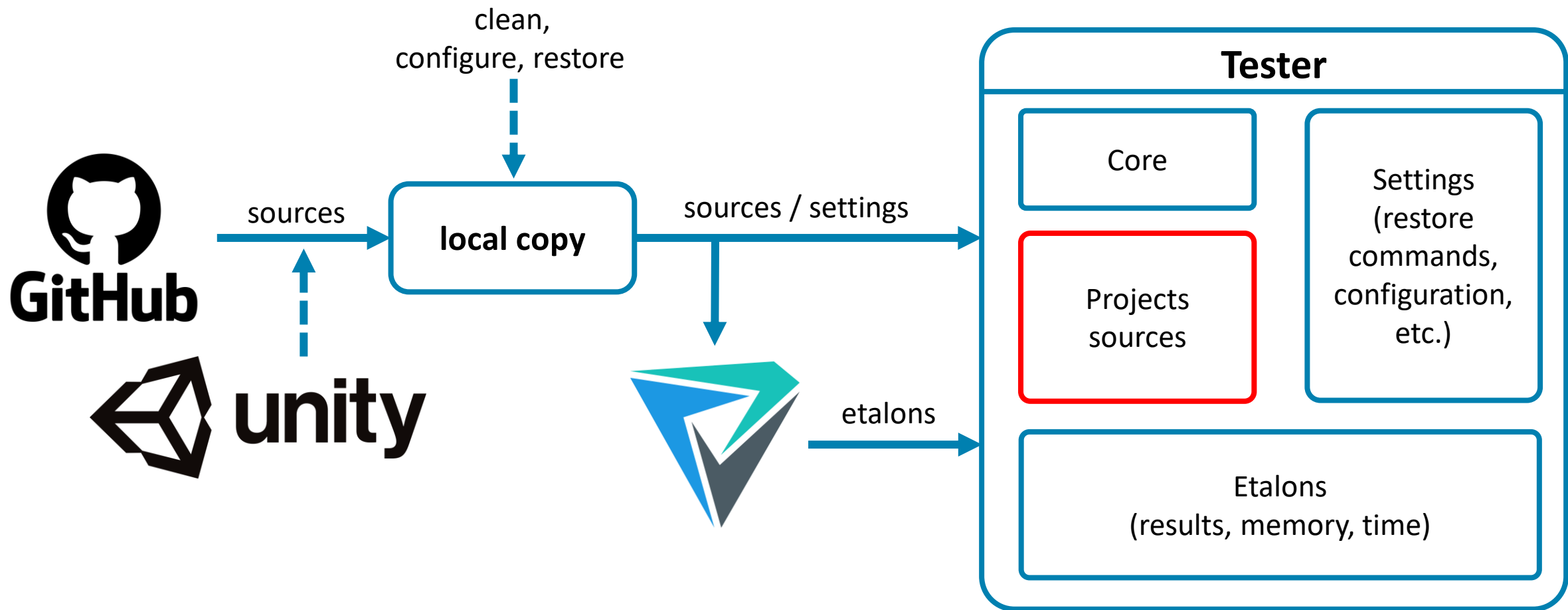
Тестирование на OSS: как?



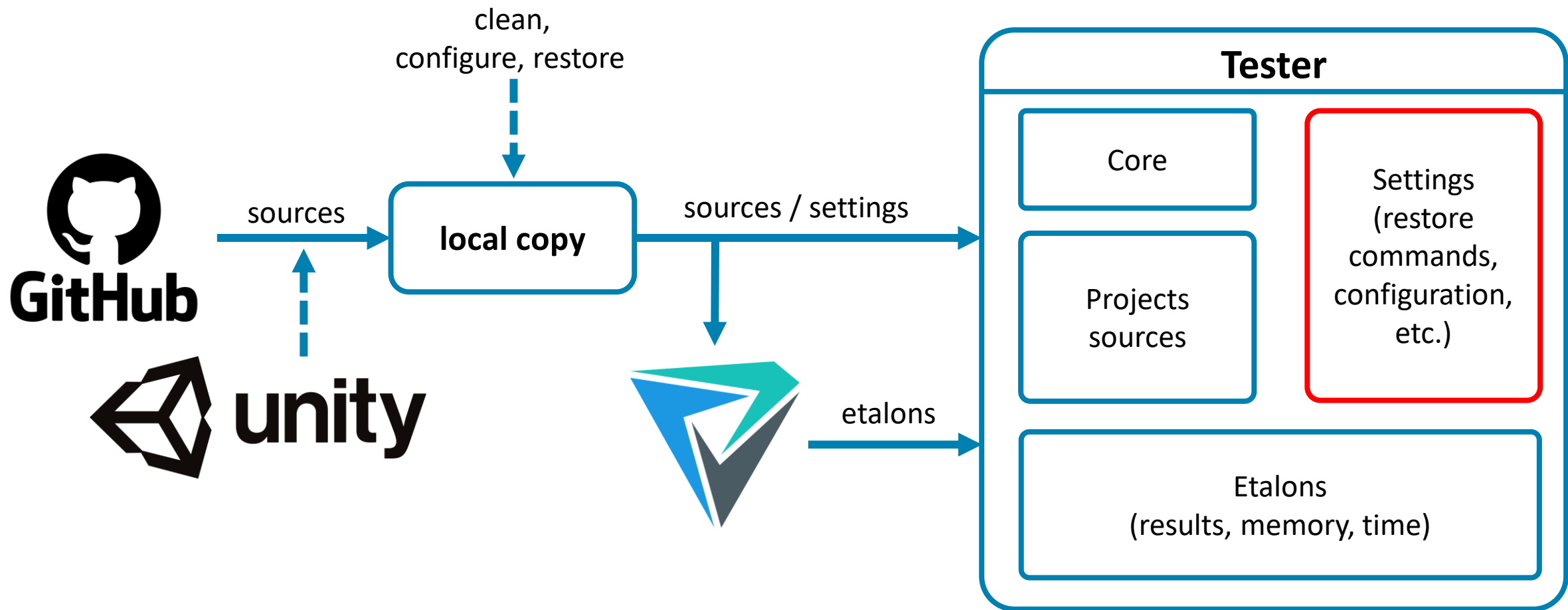
Тестирование на OSS: как?



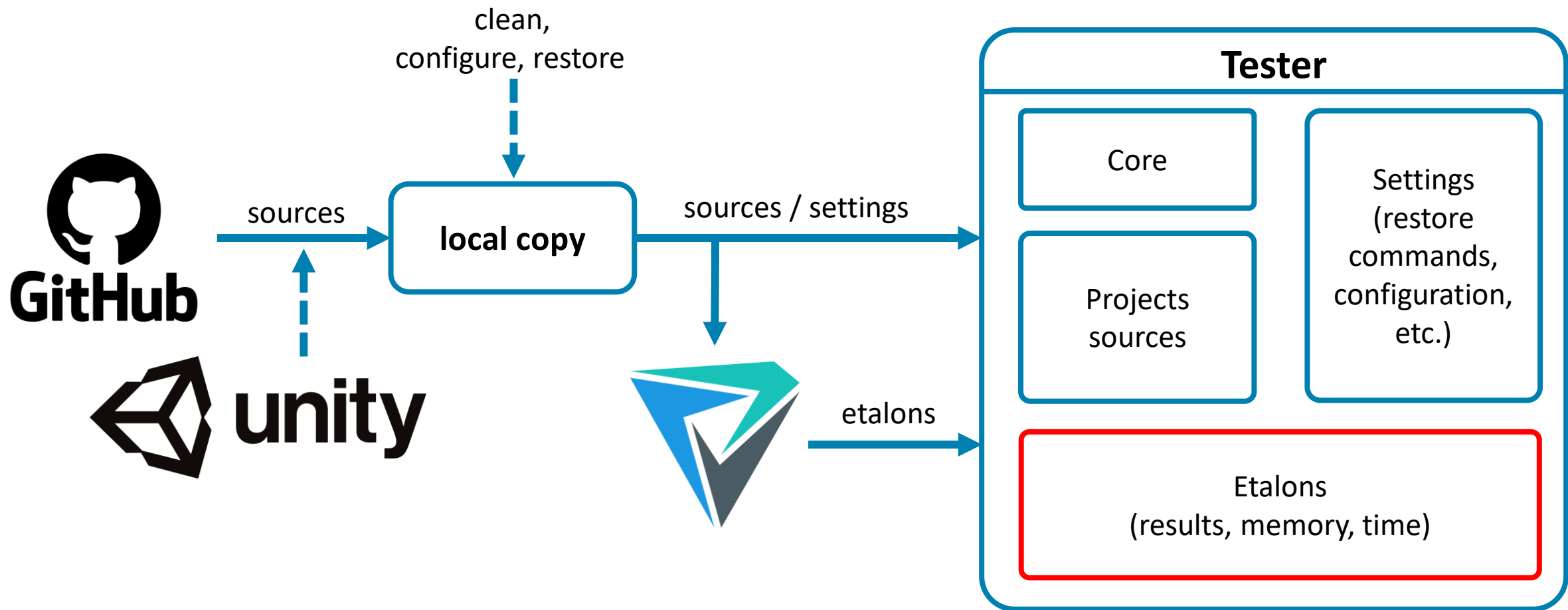
Тестирование на OSS: как?



Тестирование на OSS: как?



Тестирование на OSS: как?



1/78	XXETests	00:00:11	(+ 0s)	0.7GiB	(+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB	(- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB	(+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB	(+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB	(+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB	(+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB	(- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB	(- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB	(+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB	(- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB	(- 0.1GiB)	Ok
....						
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB	(+ 0.4GiB)	Diff
....						
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB	(- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB	(- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB	(+ 0.0GiB)	Ok

1/78	XXETests	00:00:11	(+ 0s)	0.7GiB	(+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB	(- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB	(+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB	(+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB	(+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB	(+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB	(- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB	(- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB	(+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB	(- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB	(- 0.1GiB)	Ok
....						
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB	(+ 0.4GiB)	Diff
....						
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB	(- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB	(- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB	(+ 0.0GiB)	Ok

1/78	XXETests	00:00:11 (+ 0s)	0.7GiB (+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11 (+12s)	1.8GiB (- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13 (+ 3s)	0.3GiB (+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11 (- 1s)	0.3GiB (+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09 (+ 2s)	0.3GiB (+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45 (+ 7s)	2.1GiB (+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39 (+ 8s)	1.3GiB (- 0.1GiB)	Diff
8/78	Npgsql	00:00:58 (+ 7s)	1.2GiB (- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09 (+ 2s)	0.2GiB (+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18 (+ 6s)	0.9GiB (- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18 (+ 6s)	0.4GiB (- 0.1GiB)	Ok
....				
24/78	SharpDevelop	00:12:30 (+ 3m)	6.4GiB (+ 0.4GiB)	Diff
....				
27/78	DotNetOpenAuth	00:01:09 (+19s)	2.3GiB (- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04 (+16s)	0.9GiB (- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40 (+16s)	1.0GiB (+ 0.0GiB)	Ok

1/78	XXETests	00:00:11	(+ 0s)	0.7GiB (+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB (- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB (+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB (+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB (+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB (+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB (- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB (- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB (+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB (- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB (- 0.1GiB)	Ok
....					
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB (+ 0.4GiB)	Diff
....					
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB (- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB (- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB (+ 0.0GiB)	Ok

1/78	XXETests	00:00:11	(+ 0s)	0.7GiB	(+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB	(- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB	(+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB	(+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB	(+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB	(+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB	(- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB	(- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB	(+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB	(- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB	(- 0.1GiB)	Ok
....						
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB	(+ 0.4GiB)	Diff
....						
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB	(- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB	(- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB	(+ 0.0GiB)	Ok

1/78	XXETests	00:00:11	(+ 0s)	0.7GiB	(+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB	(- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB	(+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB	(+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB	(+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB	(+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB	(- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB	(- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB	(+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB	(- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB	(- 0.1GiB)	Ok
....						
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB	(+ 0.4GiB)	Diff
....						
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB	(- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB	(- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB	(+ 0.0GiB)	Ok



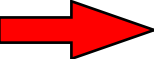
1/78	XXETests	00:00:11	(+ 0s)	0.7GiB	(+ 0.1GiB)	Ok
2/78	ComplexTransitiveDeps	00:01:11	(+12s)	1.8GiB	(- 0.1GiB)	Ok
3/78	TestTFMResolverProject	00:00:13	(+ 3s)	0.3GiB	(+ 0.0GiB)	Ok
4/78	NoMetadataRefs	00:00:11	(- 1s)	0.3GiB	(+ 0.0GiB)	Ok
5/78	TFSTestProject	00:00:09	(+ 2s)	0.3GiB	(+ 0.0GiB)	Ok
6/78	CoreWF	00:01:45	(+ 7s)	2.1GiB	(+ 0.1GiB)	Diff
7/78	Cofoundry	00:00:39	(+ 8s)	1.3GiB	(- 0.1GiB)	Diff
8/78	Npgsql	00:00:58	(+ 7s)	1.2GiB	(- 0.2GiB)	Ok
9/78	ConfigTestCSharp	00:00:09	(+ 2s)	0.2GiB	(+ 0.0GiB)	Ok
10/78	VS2017Test	00:00:18	(+ 6s)	0.9GiB	(- 0.2GiB)	Diff
11/78	VS2019Test	00:00:18	(+ 6s)	0.4GiB	(- 0.1GiB)	Ok
....						
24/78	SharpDevelop	00:12:30	(+ 3m)	6.4GiB	(+ 0.4GiB)	Diff
....						
27/78	DotNetOpenAuth	00:01:09	(+19s)	2.3GiB	(- 0.7GiB)	Fail
28/78	Hearthstone Deck Tracker	00:01:04	(+16s)	0.9GiB	(- 0.1GiB)	Ok
29/78	IdentityServer3	00:00:40	(+16s)	1.0GiB	(+ 0.0GiB)	Ok



BouncyCastle C#

```
public static string ToString(object[] a)
```

```
{
```

```
     StringBuilder sb = new StringBuilder('[');
```

```
    if (a.Length > 0)
```

```
    {
```

```
        sb.Append(a[0]);
```

```
        for (int index = 1;
```

```
            index < a.Length; ++index)
```

```
        {
```

```
            sb.Append(", ").Append(a[index]);
```

```
        }
```

```
    }
```

```
    sb.Append(']');
```

```
    return sb.ToString();
```

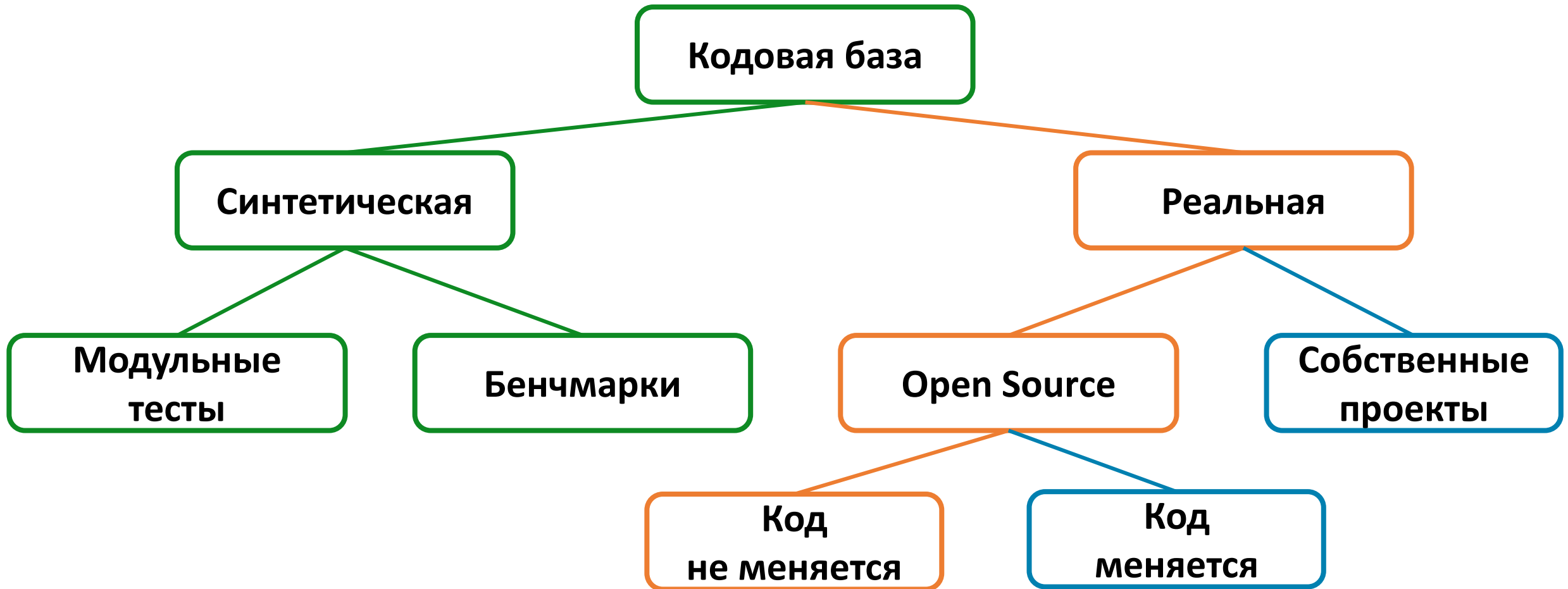
```
}
```

public StringBuilder(int capacity)

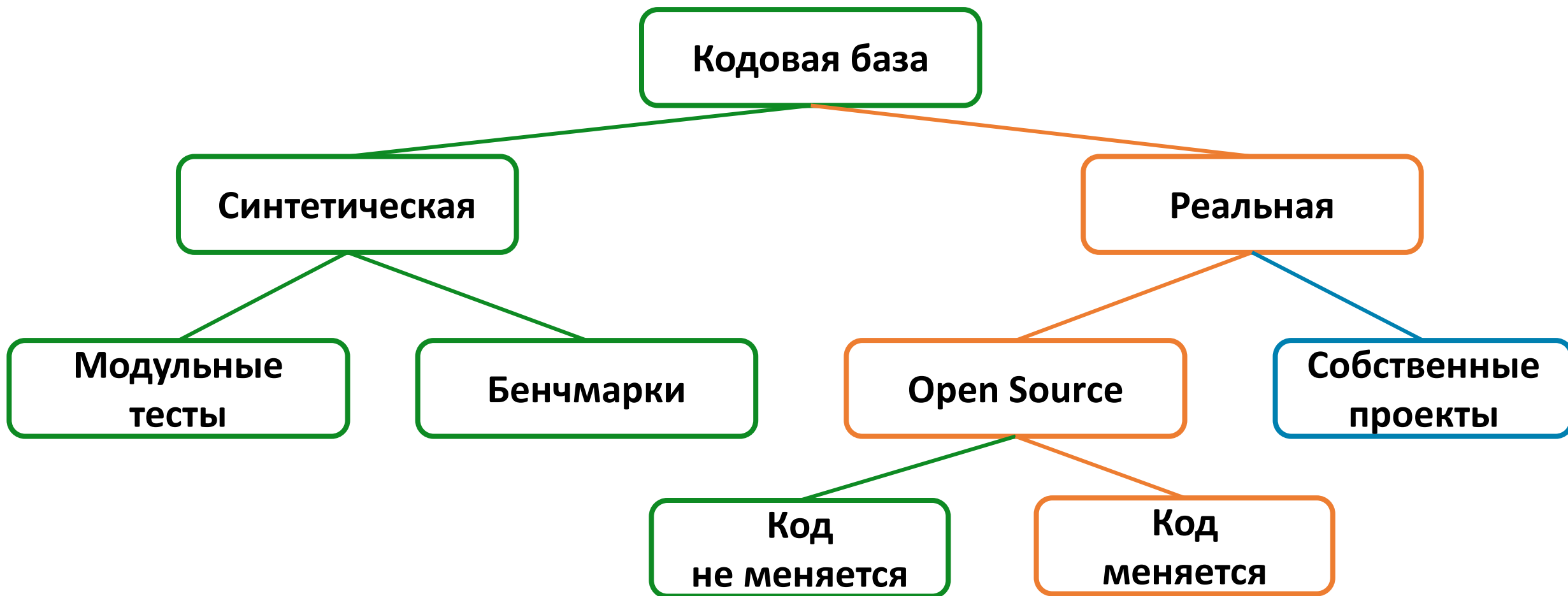


Тестирование на OSS близко
к реальному использованию

На чём тестировать?



На чём тестировать?



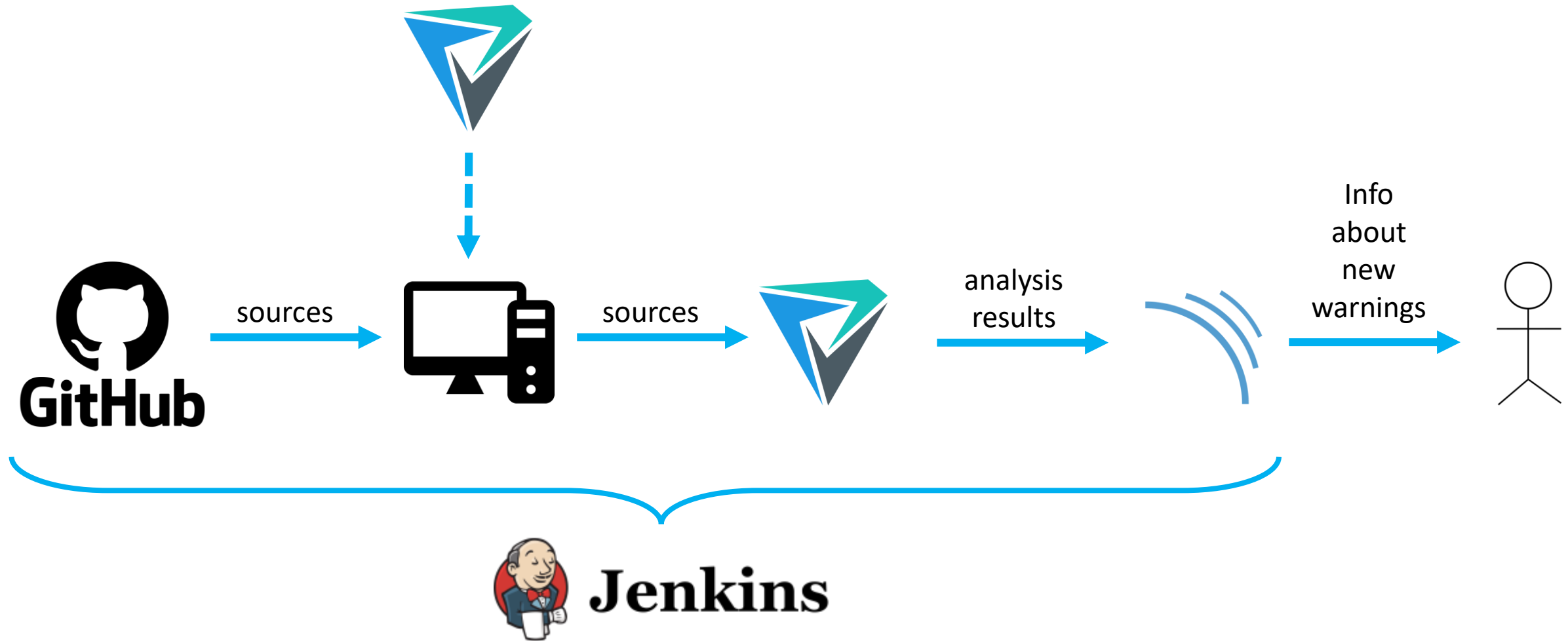
PVS-Studio & SonarQube & Open Source

Зачем?

- Тестирование ещё ближе к реальным условиям
- Демонстрация пользы статического анализа



Как?



Защита от поспешных правок кода

```
668 673 if (ResetFilePos) fseek(dat, CurPos, 0);
669 674 return ret;
675 + delete[] buf;
676 + delete[] CloseNode;
677 + delete[] CloseParent;
678 }
```



#C++

24 Мар 2021

Как PVS-Studio защищает от поспешных правок кода

Андрей Карпов

Хотя только недавно была заметка про проект CovidSim, есть хороший повод вновь про него вспомнить и продемонстрировать пользу регулярного использования PVS-Studio. Бывает, что все мы спешим и вносим правки в код, потеряв сосредоточенность. Статический анализатор может...

...

```
452
453 /* get frame to copy data into (if no frame returned, then just ignore) */
454 gpf = BKE_gpencil_layer_frame_get(gpId, gpf->framenum, GP_GETFRAME_ADD_NEW);
455 + /* Ensure to use same keyframe type. */
456 + gpf->key_type = gpf->key_type;
457 if (gpf) {
458     bGPDstroke *gps, *gpsn;
459 }
```



#C++

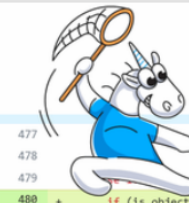
18 Янв 2022

Как PVS-Studio защищает от поспешных правок кода, пример N2

Андрей Карпов

Большое количество ошибок программистами допускается просто по невнимательности или из-за спешки. Хорошо это видно на небольших неправильных изменениях, вносимых в код. Рассмотрим как раз такой случай, когда, исправляя одну ошибку, программист добавляет новую.

...



```
477
478
479 if (is_object_active && !(base->object->mode & OB_MODE_OBJECT)) {
480 + if (is_object_active && !(base->object->mode & OB_MODE_OBJECT)) {
481 + /* Pass. Consider the selection of elements being edited. */
482 + }
483 + else if ((base->flag & BASE_SELECTED) || (base->flag_legacy & BA_MAS_SEL)) {
484     continue;
485 }
486 }
```

#C++

16 Фев 2022

Как PVS-Studio защищает от поспешных правок кода, пример N3

Андрей Карпов

Продолжаем серию маленьких заметок про то, как анализатор PVS-Studio может быстро находить новые ошибки в коде. При условии, конечно, что он регулярно используется :). Итак, перед нами очередной баг в проекте Blender.

...

```
83 + template<typename T, int Size>
84 + inline vec_base<T, Size> clamp(const vec_base<T, Size> &a, const T &min, const
85 {
86 + vec_base<T, Size> result = a;
87 + for (int i = 0; i < Size; i++) {
88 +     std::clamp(result[i], min, max);
89 + }
90 + return result;
91 }
```



#C++

18 Фев 2022

Как PVS-Studio защищает от поспешных правок кода, пример N4

Андрей Карпов

Если регулярно использовать статический анализатор кода, то можно сократить время на гадание, почему новый код работает как-то не так, как задумывалось. Рассмотрим очередную интересную ошибку, когда в процессе рефакторинга сломалась функция и это осталось не замеченным...

...

Example №1

Blender

```
#define CLAMP(a, b, c) \
{ \
    if ((a) < (b)) { \
        (a) = (b); \
    } \
    else if ((a) > (c)) { \
        (a) = (c); \
    } \
} \
(void)0
```

```
template <typename T> inline T
clamp(const T &a, const bT &min_v, const bT &max_v)
{
    T result = a;
    for (int i = 0; i < T::type_length; i++) {
        CLAMP(result[i], min_v, max_v);
    }
    return result;
}
```

Blender

```
#define CLAMP(a, b, c) \
{ \
    if ((a) < (b)) { \
        (a) = (b); \
    } \
    else if ((a) > (c)) { \
        (a) = (c); \
    } \
} \
(void)0
```

[MIN_VAL >= CUR_VAL <= MAX_VAL]

```
template <typename T> inline T
clamp(const T &a, const bT &min_v, const bT &max_v)
{
    T result = a;
    for (int i = 0; i < T::type_length; i++) {
        CLAMP(result[i], min_v, max_v);
    }
    return result;
}
```

Blender

```
#define CLAMP(a, b, c) \
{ \
    if ((a) < (b)) { \
        (a) = (b); \
    } \
    else if ((a) > (c)) { \
        (a) = (c); \
    } \
} \
(void)0
```

[MIN_VAL >= CUR_VAL <= MAX_VAL]

```
template <typename T> inline T
clamp(const T &a, const bT &min_v, const bT &max_v)
{
    T result = a;
    for (int i = 0; i < T::type_length; i++) {
        CLAMP(result[i], min_v, max_v);
    }
    return result;
}
```

Blender

```
template <typename T, int Size>
inline vec_base<T, Size>
clamp(const vec_base<T, Size> &a, const T &min, const T &max)
{
    vec_base<T, Size> result = a;
    for (int i = 0; i < Size; i++) {
        std::clamp(result[i], min, max);
    }
    return result;
}
```

Blender

```
template <typename T, int Size>
inline vec_base<T, Size>
clamp(const vec_base<T, Size> &a, const T &min, const T &max)
{
    vec_base<T, Size> result = a;
    for (int i = 0; i < Size; i++) {
        → std::clamp(result[i], min, max);
    }
    return result;
}
```

Blender

```
template <typename T, int Size>
inline vec_base<T, Size>
clamp(const vec_base<T, Size> &a, const T &min, const T &max)
{
    vec_base<T, Size> result = a;
    for (int i = 0; i < Size; i++) {
        result[i] = std::clamp(result[i], min, max);
    }
    return result;
}
```

Example №2

COVID-19 CovidSim Model

```
int GetXMLNode(...)\n{\n    char buf[65536], CloseNode[2048], CloseParent[2048];\n    ....\n    if (ResetFilePos) fseek(dat, CurPos, 0);\n    return ret;\n}
```


COVID-19 CovidSim Model

```
int GetXMLNode(...)\n{\n    char buf[65536], CloseNode[2048], CloseParent[2048];\n    ....\n    if (ResetFilePos) fseek(dat, CurPos, 0);\n    return ret;\n}
```

COVID-19 CovidSim Model

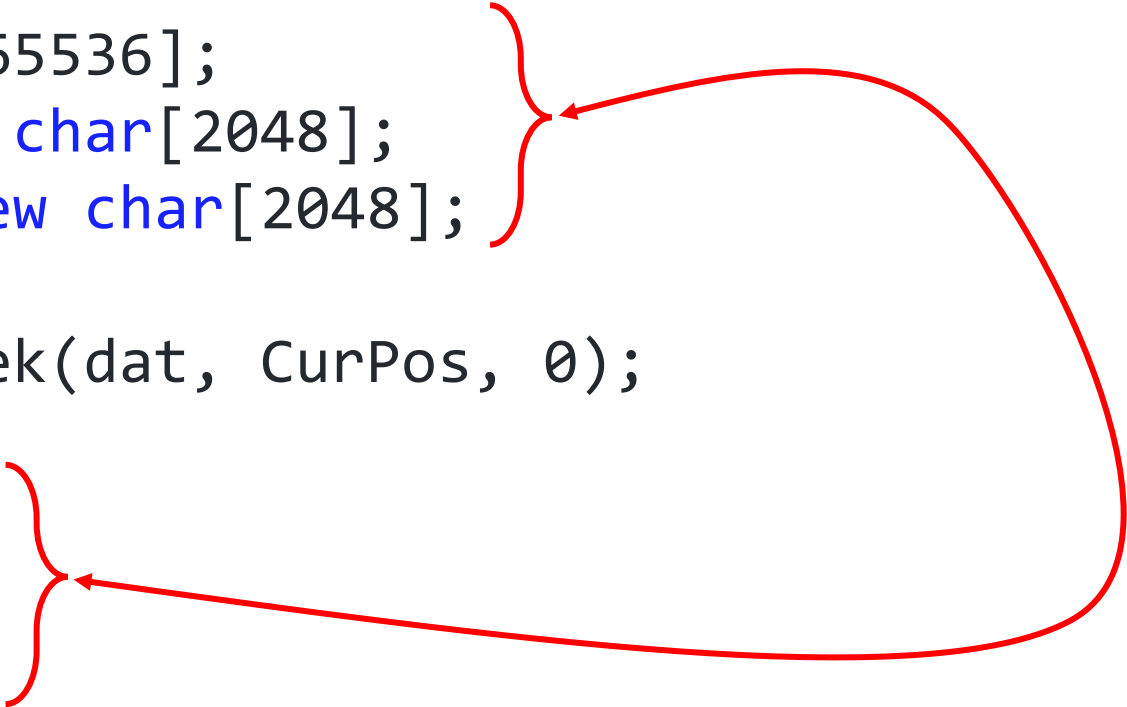
```
int GetXMLNode(...)  
{  
    char buf[65536], CloseNode[2048], CloseParent[2048];  
    ....  
    if (ResetFilePos) fseek(dat, CurPos, 0);  
    return ret;  
}
```

COVID-19 CovidSim Model

```
int GetXMLNode(...)  
{  
    char* buf = new char[65536];  
    char* CloseNode = new char[2048];  
    char* CloseParent = new char[2048];  
    ....  
    if (ResetFilePos) fseek(dat, CurPos, 0);  
    return ret;  
    delete[] buf;  
    delete[] CloseNode;  
    delete[] CloseParent;  
}
```

COVID-19 CovidSim Model

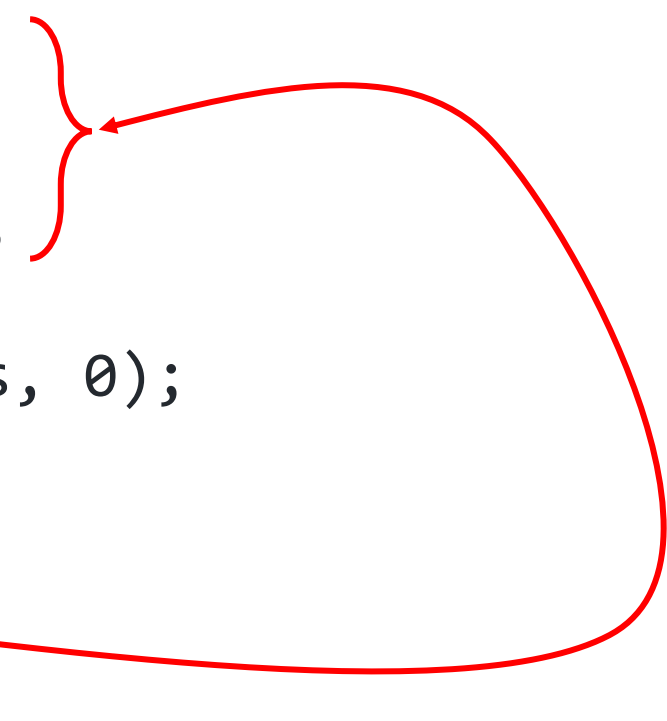
```
int GetXMLNode(...)  
{  
    char* buf = new char[65536];  
    char* CloseNode = new char[2048];  
    char* CloseParent = new char[2048];  
    ....  
    if (ResetFilePos) fseek(dat, CurPos, 0);  
    return ret;  
    delete[] buf;  
    delete[] CloseNode;  
    delete[] CloseParent;  
}
```



A red curved arrow originates from the right side of the first group of three new char array declarations (buf, CloseNode, CloseParent) and points to the right side of the three corresponding delete[] statements, indicating that these buffers are deallocated after use.

COVID-19 CovidSim Model

```
int GetXMLNode(...)\n{\n    char* buf = new char[65536];\n    char* CloseNode = new char[2048];\n    char* CloseParent = new char[2048];\n    ....\n    if (ResetFilePos) fseek(dat, CurPos, 0);\n    return ret;\n    delete[] buf;\n    delete[] CloseNode;\n    delete[] CloseParent;\n}
```



A red arrow points from the left to the `return ret;` line. Two red curly braces are present: one on the right grouping the three `new char` allocation lines, and another on the left grouping the three `delete[]` deallocation lines. A red curved arrow originates from the right brace and points to the left brace, indicating the flow of memory deallocation after the function returns.

Example №3

Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                       gpfs->framenum,
                                       GP_GETFRAME_ADD_NEW);

    if (gpf) {
        ....
    }
    ....
}
```

Blender

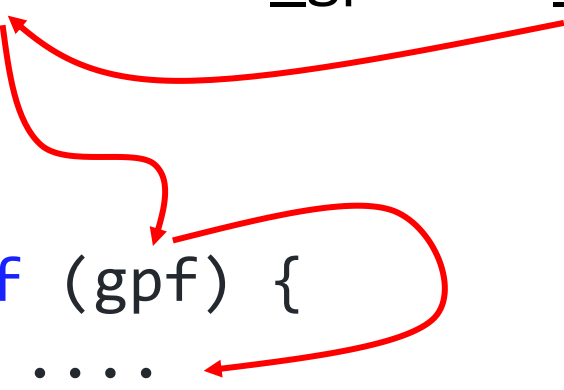
```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                     gpfs->framenum,
                                     GP_GETFRAME_ADD_NEW);

    if (gpf) {
        ....
    }
    ....
}
```



Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);
    if (gpf) {
        ....
    }
    ....
}
```



Red arrows highlight the variable `gpf` and its use in the `if` statement. One arrow points from the `gpf` variable in the assignment to the `gpf` in the `if` condition. Another arrow points from the `gpfs->framenum` argument to the `gpfs` variable, and a third arrow points from the `GP_GETFRAME_ADD_NEW` argument to the `GP` part of the constant.

Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);

    gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```

Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);
    → gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```

Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                     gpfs->framenum,
                                     GP_GETFRAME_ADD_NEW);

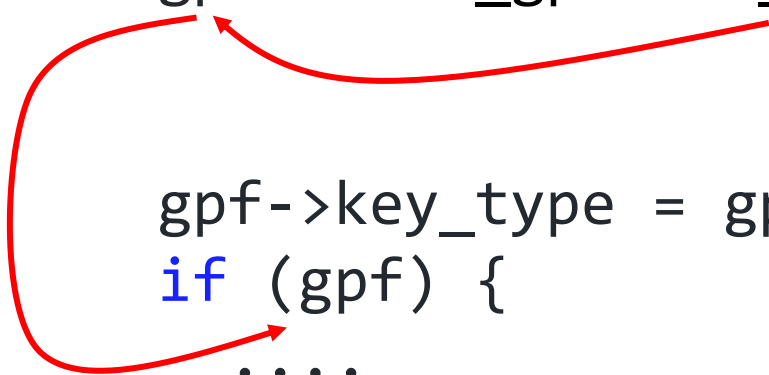
    gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```



Blender

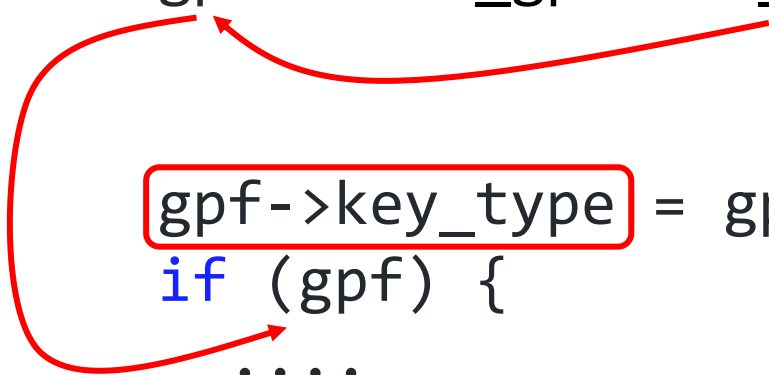
```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);

    gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```



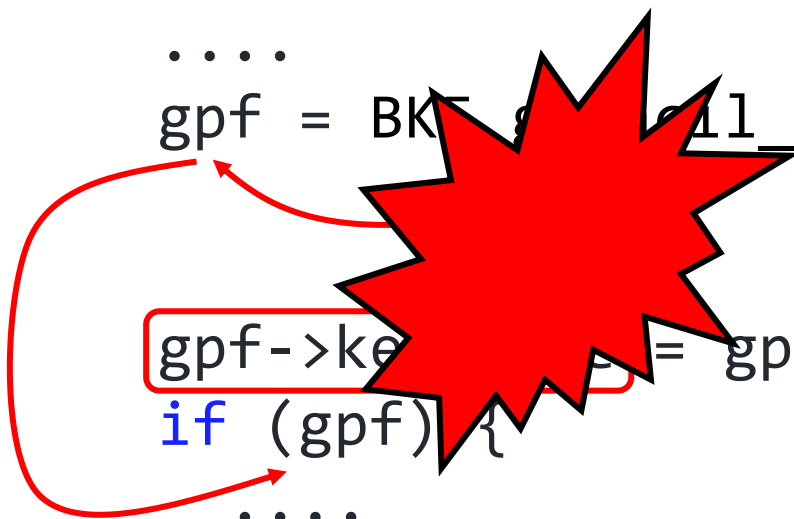
Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);
    gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```



Blender

```
bool ED_gpencil_anim_copybuf_paste(....)
{
    ....
    gpf = BKE_gpencil_layer_frame_get(gpId,
                                      gpfs->framenum,
                                      GP_GETFRAME_ADD_NEW);
    gpf->key_type = gpfs->key_type;
    if (gpf) {
        ....
    }
    ....
}
```



Example №4

Akka.NET

```
case ShardId shardId:
    _shards.Add(shardId);
    return true;
case SnapshotOffer offer when (offer.Snapshot is ShardCoordinator.CoordinatorState state):
    _shards.UnionWith(state.Shards.Keys.Union(state.UnallocatedShards));
    return true;
case SnapshotOffer offer when (offer.Snapshot is State state):
    _shards.Union(state.Shards);
    _writtenMarker = state.WrittenMigrationMarker;
    return true;
case RecoveryCompleted _:
    Log.Debug("Recovery complete. Current shards [{0}]. Written Marker {1}",
        string.Join(", ", _shards), _writtenMarker);

    if (!_writtenMarker) {
        Persist(MigrationMarker.Instance, _ => {
            Log.Debug("Written migration marker");
            _writtenMarker = true;
        });
    }
    return true;
case MigrationMarker _:
    _writtenMarker = true;
    return true;
```

Akka.NET

```
case ....:
    _shards.Add(shardId);
    ....
case ....:
    _shards.UnionWith(state.Shards.Keys.Union(state.UnallocatedShards));
    ....
case ....:
    _shards.Union(state.Shards);
    ....

// _shards: HashSet<T>
```



VasilievSerg commented on Apr 21



Hi,
I found the suspicious code via the PVS-Studio analyzer.

Here is the code:

```
...
case ShardId shardId:
    _shards.Add(shardId); // <=
    return true;
case SnapshotOffer offer when (offer.Snapshot is ShardCoordinator.CoordinatorState state):
    _shards.UnionWith(state.Shards.Keys.Union(state.UnallocatedShards)); // <=
    return true;
case SnapshotOffer offer when (offer.Snapshot is State state):
    _shards.Union(state.Shards); // <=
...
```

Here is [the link](#) to the sources.

Several methods were called for the `_shards` variable. The last invocation looks suspicious.

`Add` and `UnionWith` are methods of the `HashSet<T>` type. They change the `_shards` object state. But `Union` is the extension method which does not change the `_shards` state. The result of this invocation is not used as well.



Aaron Stannard @Aaronontheweb · Apr 25



Replying to [@_SergVasiliev_](#) and [@AkkaDotNET](#)

I really appreciated this issue being opened on the repo too because it was extremely subtle but important to how a complex part of [@AkkaDotNET](#) works. That change was introduced in a big PR that touched 100+ files and I spent more than 10 hours reviewing it.



1



1



Aaron Stannard @Aaronontheweb · Apr 25



Replying to [@Aaronontheweb](#) [@_SergVasiliev_](#) and [@AkkaDotNET](#)

I probably wouldn't have caught that error without:

1. A user writing a detailed bug report about an intermittent issue that is very hard to replicate and us spending tens of hours repro-ing it after this was released

or

2. Your static analysis tool



1



1



2

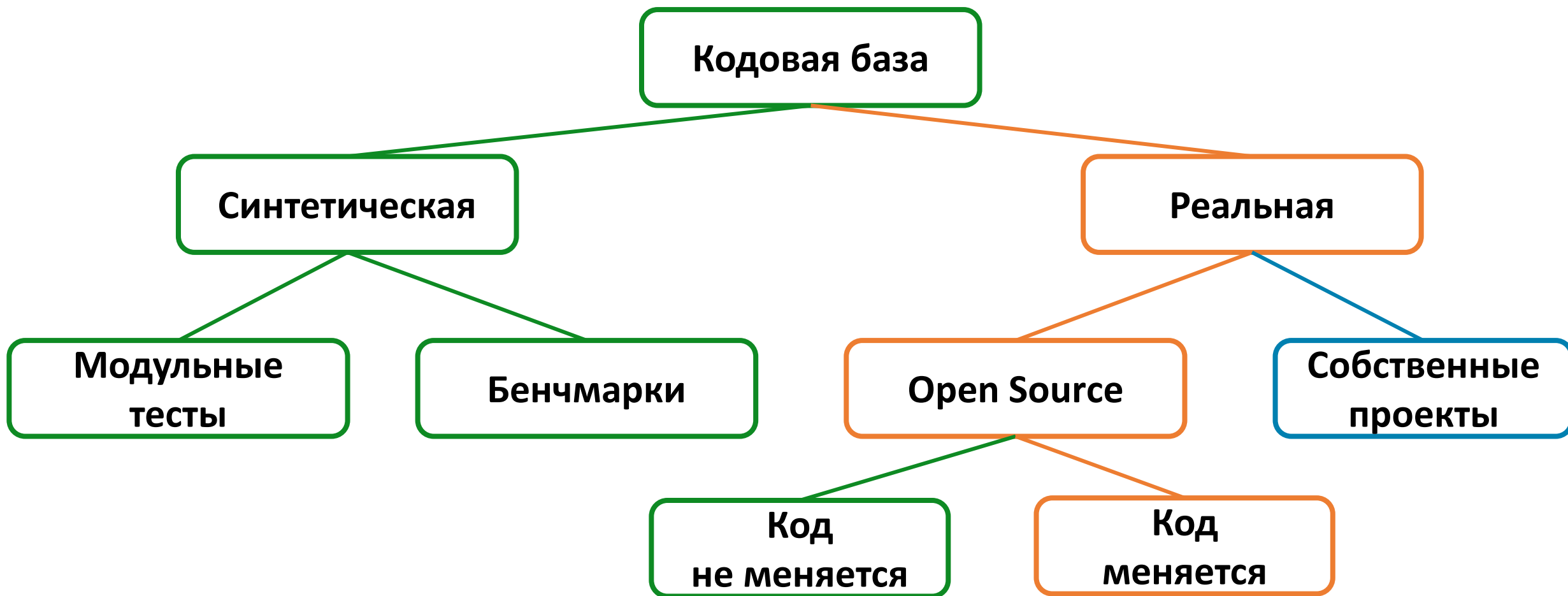


Akka.NET

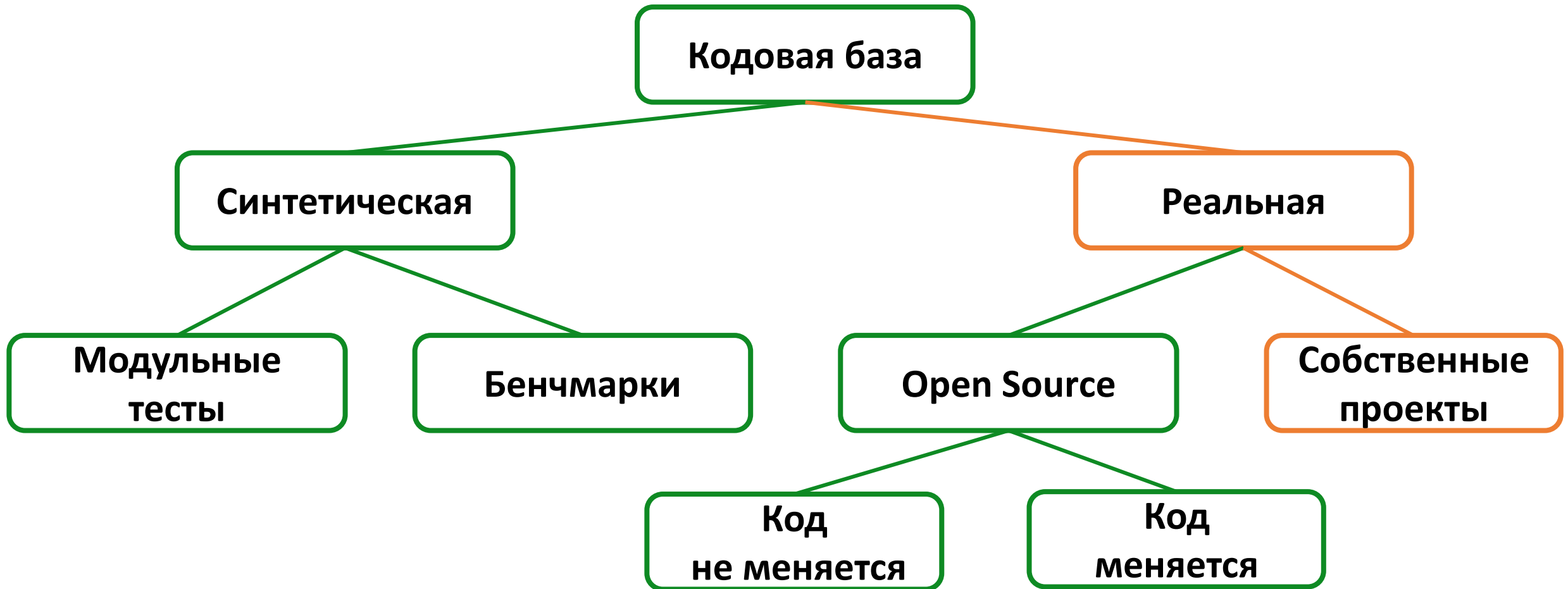
```
case ShardId shardId:
    _shards.Add(shardId);
    return true;
case SnapshotOffer offer when (offer.Snapshot is ShardCoordinator.CoordinatorState state):
    _shards.UnionWith(state.Shards.Keys.Union(state.UnallocatedShards));
    return true;
case SnapshotOffer offer when (offer.Snapshot is State state):
    _shards.Union(state.Shards);
    _writtenMarker = state.WrittenMigrationMarker;
    return true;
case RecoveryCompleted _:
    Log.Debug("Recovery complete. Current shards [{0}]. Written Marker {1}",
        string.Join(", ", _shards), _writtenMarker);

    if (!_writtenMarker) {
        Persist(MigrationMarker.Instance, _ => {
            Log.Debug("Written migration marker");
            _writtenMarker = true;
        });
    }
    return true;
case MigrationMarker _:
    _writtenMarker = true;
    return true;
```

На чём тестировать?



На чём тестировать?



Static analyzer vs own sources



А вы проверяете PVS-Studio
с помощью PVS-Studio?



PVS-Studio vs PVS-Studio

```
public override void  
VisitAnonymousObjectCreationExpression(  
    AnonymousObjectCreationExpressionSyntax node)  
{  
    foreach (var initializer in node?.Initializers)  
        initializer?.Expression?.Accept(this);  
}
```



PVS-Studio vs PVS-Studio

```
public override void  
VisitAnonymousObjectCreationExpression(  
    AnonymousObjectCreationExpressionSyntax node)  
{  
    foreach (var initializer in node?.Initializers)  
        initializer?.Expression?.Accept(this);  
}
```



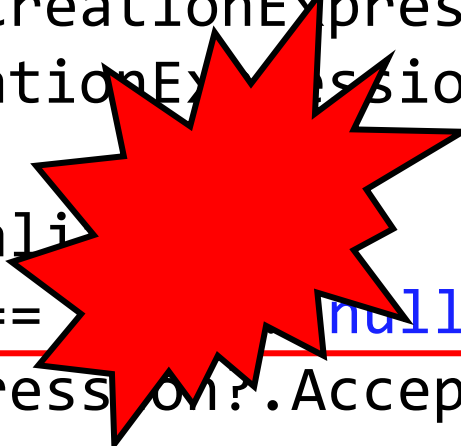
PVS-Studio vs PVS-Studio

```
public override void  
VisitAnonymousObjectCreationExpression(  
    AnonymousObjectCreationExpressionSyntax node)  
{  
    foreach (var initializer  
        in (node == null ? null : node.Initializers))  
        initializer?.Expression?.Accept(this);  
}
```



PVS-Studio vs PVS-Studio

```
public override void  
VisitAnonymousObjectCreationExpression(  
    AnonymousObjectCreationExpressionSyntax node)  
{  
    foreach (var initializer  
             in (node == null : node.Initializers))  
        initializer?.Expression?.Accept(this);  
}
```



PVS-Studio vs PVS-Studio



#CSharp

03 Июн 2021

Использование оператора ?. в
foreach: защита от
NullReferenceException, которая не
работает

Сергей Васильев

Любите оператор '?.'? А кто же не любит? Эти лаконичные
проверки на null нравятся многим. Однако сегодня мы
поговорим о случае, когда оператор ?. только создаёт
иллюзию безопасности. Речь пойдёт о его использовании в
цикле foreach.

...



NETHERMIND®

PVS-Studio vs PVS-Studio

```
public bool GeneratePreprocessedFile(...) {  
    ....  
    if (info.PreprocessorCommandLine.Contains(" /arch:SSE"))  
        ClangCommandLine += " /D \"_M_IX86_FP=1\"";  
    else if (info.PreprocessorCommandLine.Contains(" /arch:SSE2"))  
        ClangCommandLine += " /D \"_M_IX86_FP=2\"";  
    else if (info.PreprocessorCommandLine.Contains(" /arch:IA32"))  
        ClangCommandLine += " /U \"_M_IX86_FP\"";  
    else if (info.PreprocessorCommandLine.Contains(" /arch:AVX"))  
        ClangCommandLine += " /D \"_M_IX86_FP=2\"";  
    ....  
}
```



PVS-Studio vs PVS-Studio

```
if (info.PreprocessorCommandLine.Contains(" /arch:SSE"))  
    ClangCommandLine += " /D \"_M_IX86_FP=1\"";  
else if (info.PreprocessorCommandLine.Contains(" /arch:SSE2"))  
    ClangCommandLine += " /D \"_M_IX86_FP=2\"";
```



PVS-Studio vs PVS-Studio

```
if (info.PreprocessorCommandLine.Contains(" /arch:SSE"))  
    ClangCommandLine += " /D \"_M_IX86_FP=1\"";  
else if (info.PreprocessorCommandLine.Contains(" /arch:SSE2"))  
    ClangCommandLine += " /D \"_M_IX86_FP=2\"";
```



PVS-Studio vs PVS-Studio

```
if (info.PreprocessorCommandLine.Contains(" /arch:SSE"))  
    ClangCommandLine += " /D \"_M_IX86_FP=1\"";  
else if (info.PreprocessorCommandLine.Contains(" /arch:SSE2"))  
    ClangCommandLine += " /D \"_M_IX86_FP=2\"";
```

Expression is always false
/ unreachable statement



PVS-Studio vs PVS-Studio

```
public void ProcessFiles(....) {  
    ....  
    int RowCount = DynamicErrorListControl.Instance  
                                     .Plog.NumberOfRows;  
  
    if (RowCount > 20000)  
        DatatableUpdateInterval = 30000; //30s  
    else if (RowCount > 100000)  
        DatatableUpdateInterval = 60000; //1min  
    else if (RowCount > 200000)  
        DatatableUpdateInterval = 120000; //2min  
    ....  
}
```



PVS-Studio vs PVS-Studio

```
public void ProcessFiles(....) {  
    ....  
    int RowCount = DynamicErrorListControl.Instance  
                                     .Plog.NumberOfRows;  
    if (RowCount > 20000)  
        DatatableUpdateInterval = 30000; //30s  
    else if (RowCount > 100000)  
        DatatableUpdateInterval = 60000; //1min  
    else if (RowCount > 200000)  
        DatatableUpdateInterval = 120000; //2min  
    ....  
}
```



PVS-Studio vs PVS-Studio

```
if (RowCount > 20000)
    ....
else if (RowCount > 100000)
    ....
else if (RowCount > 200000)
    ....
```

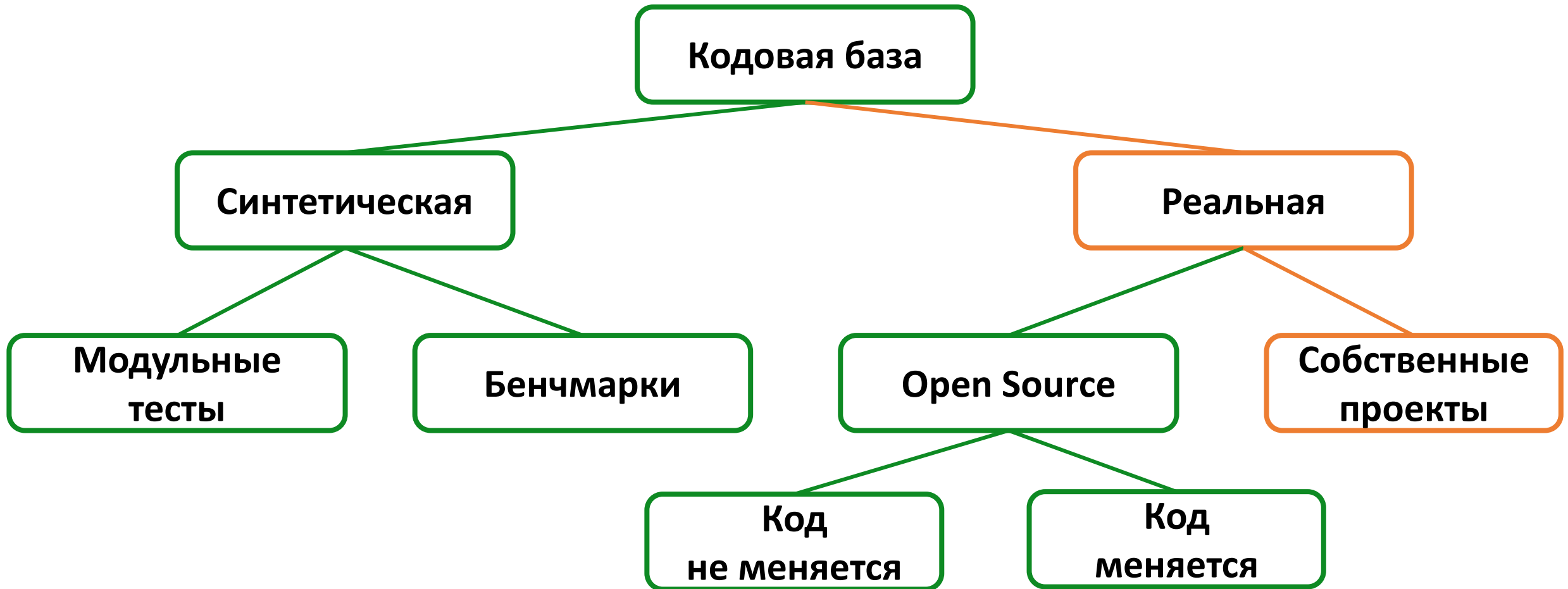


PVS-Studio vs PVS-Studio

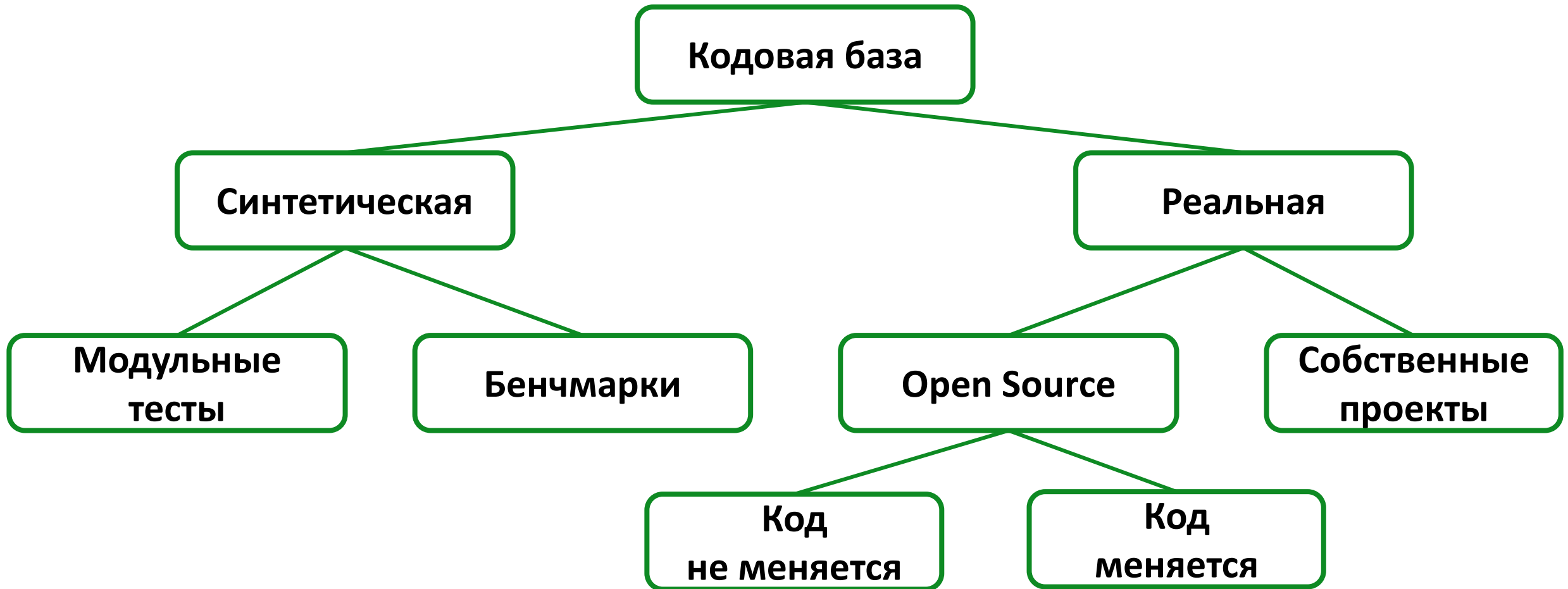
```
if (RowCount > 20000)
    ....
// RowCount <= 20 000
else if (RowCount > 100000)
    ....
else if (RowCount > 200000)
    ....
```



На чём тестировать?



На чём тестировать?





#CSharp

09 Сен 2019

Лучшее - враг хорошего

Сергей Хренов

Эта статья о том, как однажды мы решили немного улучшить внутренний инструмент SelfTester, применяемый для проверки качества работы анализатора PVS-Studio. Улучшение было несложным и выглядело полезным, но создало нам много проблем, и впоследствии выяснилось, что лучше...



07 Дек 2017

Когда дворецкий - жертва

Сергей Хренов

Сегодня мы отойдём от классического клише второсортных детективов и расскажем вам о случае из нашей практики, когда дворецкий сам выступил в роли жертвы, а поиски настоящего преступника привели нас к неожиданному результату. Но не стоит пугаться. Речь, конечно же...



@_SergVasiliev_



Сергей
Васильев

pvs-studio.com
vasiliev@viva64.com