

WinUnit

お気楽お手軽 UnitTest

わんくま同盟

MVP for Visual C++ (2004-2008)

ΕΠΙΣΤΗΜΗ

<http://blogs.wankuma.com/episteme/>
episteme@cppl.jp



επιστημη ちゃ何者ぞ!?

- C++界ではちった名の知れたソフト屋
- わんくま同盟会員番号12番
- 15年前からの物書き(DDJJ,C-Mag. etc.)
- C++標準化委員会の中のひと
- Databaseおんち
- 組み込み? Z80/8086でやりました
- わんくま同盟茶藝部顧問 ← いまここ



さて今日のお題は

WinUnit

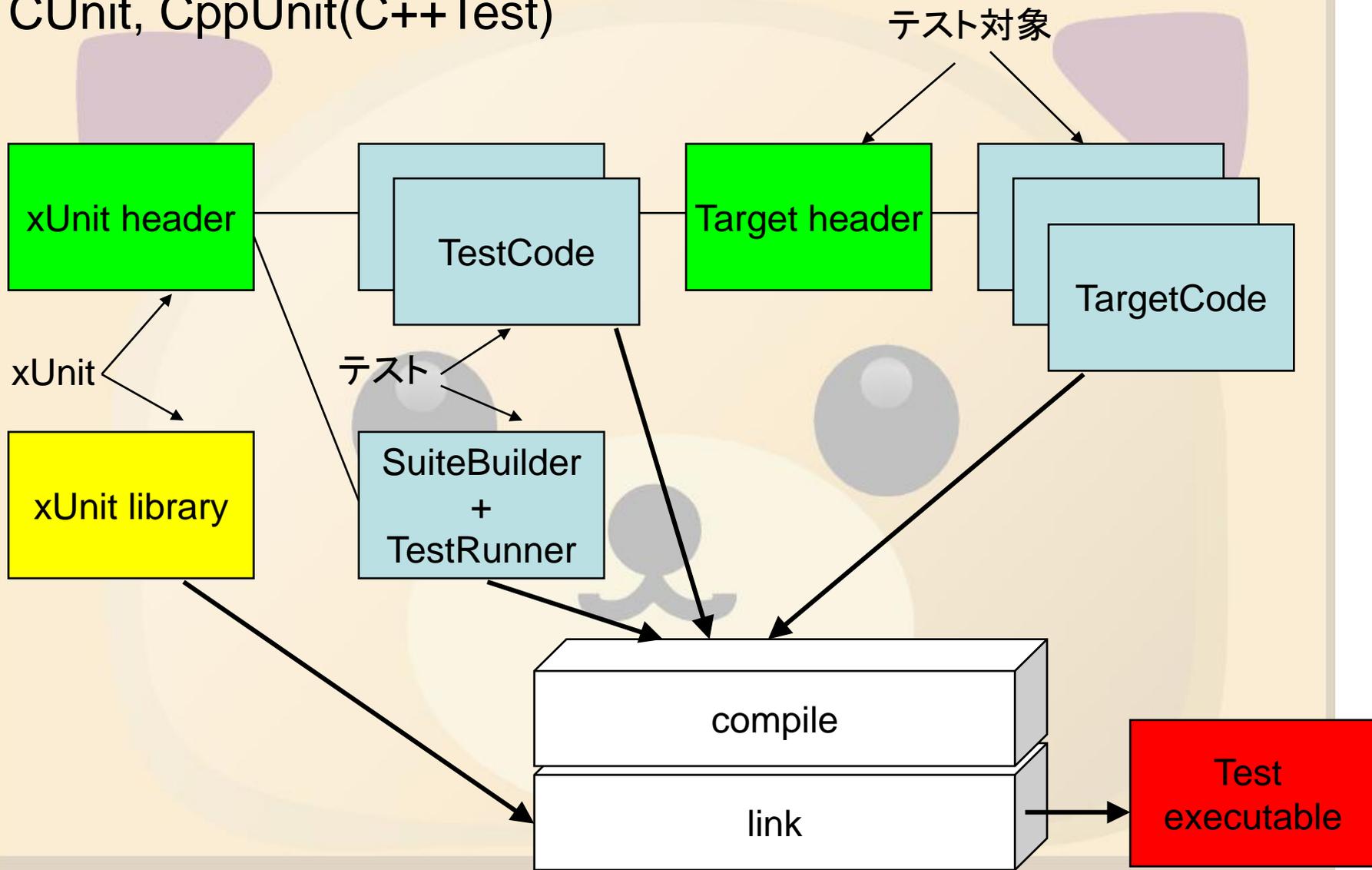


わんくま同盟 名古屋勉強会 #2

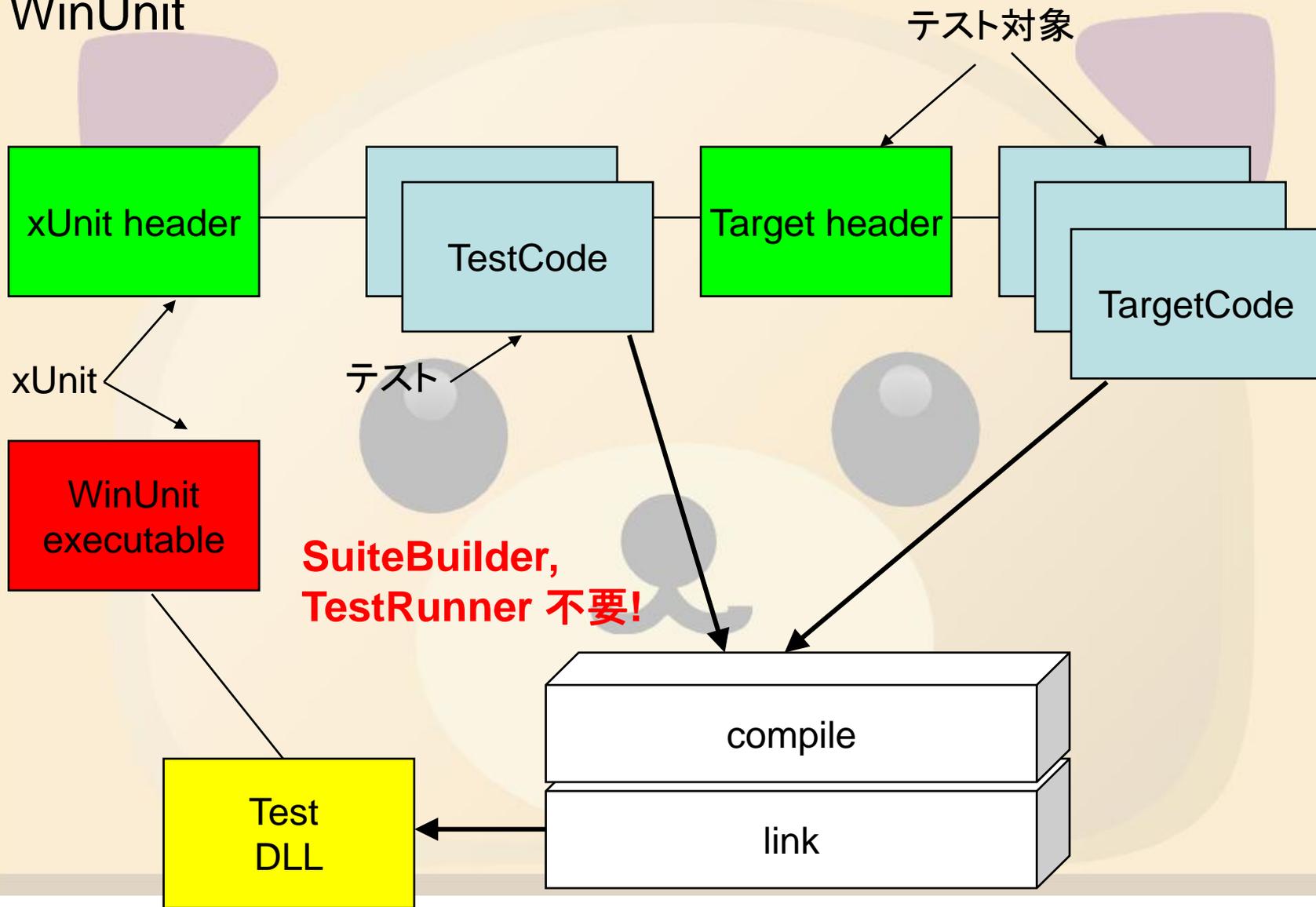
WinUnitってば

- 数あるUnitTestFrameworkのひとつ
- MSDN magazine Feb.2008 に収録
- Windows, Visual C++に**限定**
- **無償** 一処理系もタダ (VC++Expressおっけ)
- JUnit/NUnit並みに**簡単**
- **小型軽量** : header二本/exe一本 だけ
- Visual Studio – IDEに統合可

CUnit, CppUnit(C++Test)



WinUnit



テスト対象 stack.h

```
#ifndef STACK_H__
#define STACK_H__

#ifdef __cplusplus
extern "C" {
#endif

typedef struct Stack_t* Stack;

Stack stack_create();
void stack_delete(Stack stk);

void stack_push(Stack stk, int value);
int stack_size(Stack stk);
void stack_pop(Stack stk);
int stack_top(Stack stk);

#ifdef __cplusplus
}
#endif

#endif
```

こいつらが
「期待通り(仕様通り)
に動いてくれるか」
をテストする。

↓
用意した入力 に対して
期待する結果 が得られる
ことを検証する。

テストのかきかた(1) stack_test.cpp

```
#include <WinUnit.h>
#include "stack.h"
```

WinUnit利用に必要なのはこんだけ。

```
namespace {
    Stack s;
}
```

テスト対象。
(名前空間を汚染しないよう
匿名namespaceで囲む)

```
FIXTURE(stack_fixture);
```

```
SETUP(stack_fixture){
    s = stack_create();
    WIN_ASSERT_NOT_NULL(s);
}
```

各テストを
SETUP(初期化)と
TEARDOWN(後始末)で
囲む

```
TEARDOWN(stack_fixture){
    stack_delete(s);
}
```



テストのかきかた(2) stack_test.cpp

//----- 初期化

```
BEGIN_TESTF(01_initialize, stack_fixture)
{
    WIN_ASSERT_ZERO(stack_size(s));
}
END_TESTF
```

結果は0になるか?

//----- push

```
BEGIN_TESTF(02_push, stack_fixture)
{
    stack_push(s, 10);
    WIN_ASSERT_EQUAL(1, stack_size(s));
    WIN_ASSERT_EQUAL(10, stack_top(s));
    stack_push(s, 20);
    WIN_ASSERT_EQUAL(2, stack_size(s));
    WIN_ASSERT_EQUAL(20, stack_top(s));
}
END_TESTF
```

結果が期待値に
等しいか?



DEMO

- Windows Vista SP1
(XP also available)

- Microsoft Visual Studio 2008
(VS2005 also available)

Testが成功しない限り
Productを作らせない

