



WEEK 11

---

**BYOD**



# MERGE REQUESTS

- ▶ Bitte schaut, dass ihr regelmäßig fertige Teilpakete in das gemeinsame Repo merget
- ▶ Ein großer Code Dump am Ende des Projekts macht Reviews schwierig und führt zu unnötigen Mergekonflikten
- ▶ Haltet die MRs klein - pro MR nur eine Änderung



# ENDPRÄSENTATION

- ▶ 15 Minuten Vortrag, 5 Minuten Diskussion = 160 Minuten
- ▶ 90 Minuten sind für 8 Gruppen zu wenig
- ▶ Drei Möglichkeiten
  - ▶ A: Mi., 1.2. und Mi., 8.2.
  - ▶ B: Mi., 1.2. und Do., 2.2.
  - ▶ C: Supermittwoch, 8.2.



# WENIGER SCHLECHT CODEN – HEUTE: VERERBUNG IN C++

```
#include <iostream>
#include <memory>

struct T {
};

struct A {
};

struct B : public A {
    T t;
};

int main() {
    std::unique_ptr<A> a1 = std::make_unique<B>();
}
```



# WENIGER SCHLECHT CODEN – HEUTE: VERERBUNG IN C++

```
#include <iostream>
#include <vector>
#include <memory>

struct T {
    T() : v(10000) {}

    std::vector<int> v;
};

struct A {};

struct B : public A {
    T t;
};

int main() {
    std::cout << std::endl << "C-Style" << std::endl;
    A *a = new B;
    delete a;

    std::unique_ptr<A> a1 = std::make_unique<B>();

    std::shared_ptr<A> a2 = std::make_shared<B>();
}
```

```
==53702==ERROR: AddressSanitizer: new-delete-type-  
mismatch on 0x60300000efe0 in thread T0:
```

```
object passed to delete has wrong type:
```

```
size of the allocated type: 24 bytes;
```

```
size of the deallocated type: 1 bytes.
```

```
#0 0x7fd544b7b0a0 in operator delete(void*, unsigned  
long) /home/user/objdir/./gcc-6.3.0/libsanitizer/asan/  
asan_new_delete.cc:108
```

```
#1 0x4010c4 in foo() /home/user/asan.cpp:27
```

```
#2 0x40117e in main /home/user/asan.cpp:33
```

```
#3 0x7fd543e7082f in __libc_start_main (/lib/x86_64-  
linux-gnu/libc.so.6+0x2082f)
```

```
#4 0x400f48 in _start (/home/user/a.out+0x400f48)
```



# WENIGER SCHLECHT CODEN – HEUTE: VERERBUNG IN C++

```
#include <iostream>
#include <vector>
#include <memory>

struct T {
    T() : v(10000) {
        std::cout << "ctor T" << std::endl;
    }

    ~T() {
        std::cout << "dtor T" << std::endl;
    }

    std::vector<int> v;
};

struct A {
};

struct B : public A {
    T t;
};
```

```
int main() {
    std::cout << std::endl << "C-Style" << std::endl;
    A *a = new B;
    delete a;

    std::cout << std::endl << "unique_ptr" << std::endl;
    std::unique_ptr<A> a1 = std::make_unique<B>();

    std::cout << std::endl << "shared_ptr" << std::endl;
    std::shared_ptr<A> a2 = std::make_shared<B>();
}
```

```
mdp-2:tmp markus$ g++-6 asan.cpp && ./a.out
```

```
C-Style
ctor T
```

```
unique_ptr
ctor T
```

```
shared_ptr
ctor T
dtor T
```



# WENIGER SCHLECHT CODEN – HEUTE: VERERBUNG IN C++

```
#include <iostream>
#include <vector>
#include <memory>

struct T {
    T() : v(10000) {
        std::cout << "ctor T" << std::endl;
    }

    ~T() {
        std::cout << "dtor T" << std::endl;
    }

    std::vector<int> v;
};

struct A {
    virtual ~A() = default;
};

struct B : public A {
    T t;
};
```

```
int main() {
    std::cout << std::endl << "C-Style" << std::endl;
    A *a = new B;
    delete a;

    std::cout << std::endl << "unique_ptr" << std::endl;
    std::unique_ptr<A> a1 = std::make_unique<B>();

    std::cout << std::endl << "shared_ptr" << std::endl;
    std::shared_ptr<A> a2 = std::make_shared<B>();
}
```

```
mdp-2:tmp markus$ g++-6 asan.cpp && ./a.out
```

```
C-Style
ctor T
dtor T

unique_ptr
ctor T

shared_ptr
ctor T
dtor T
dtor T
```