C++ONLINE

ROTH MICHAELS

TALK:

ADVENTURES WITH LEGACY CODEBASES

TALES OF INCREMENTAL IMPROVEMENT

2025

Roth Michaels
Principal Software Engineer
Native Instruments

NATIVE INSTRUMENTS

iZOTOPE

Plugin Alliance

BRAINWORX


Inst 3
Manuell
Vergleichen
Widerrufen
Wiederholen
Ansicht:
Editor
7 SKIES, Magnificence - THE DRILL
FILTER
CUTOFF
RESO
ENV AMT
THE DRILL
FLT ENV
ATTACK
DECAY
SUSTAIN
RELEASE
AMP
LAYER 1
LAYER 2
AMP ENV
Inst 4
gr808
808 Center
Mode
[mono]
808 Sides
Panning
[attack: fast]
[releaseeeeee]
Reverb: [Cloud Grain]
Size
Amount:
gr808
kick
Rev Lo-Cut:
Rev Hi-Cut:
Presets: 1 2 3
Clay and Kelsy
Bouncy FX: [off]
Speed ModW:
Kontakt
Inst 2
MS20 Drums
Cutoff
Saturation
Reverb
Delay
Delay Time
Peter Flint:
MS20 Drums
Kontakt

You can do it!

Legacy Code

...make sure new code is better

```
template <typename Derived, typename Ps>
class HasProperties {
private:
  template <typename P>
  void createProperty() {
    auto success = m_propertyHolder.CreateProperty(
                     getName<P>(),
                     getName<typename P::property_type>(),
                     T::defaultValue);
    assert(success);
  }
};
```

Roth Michaels

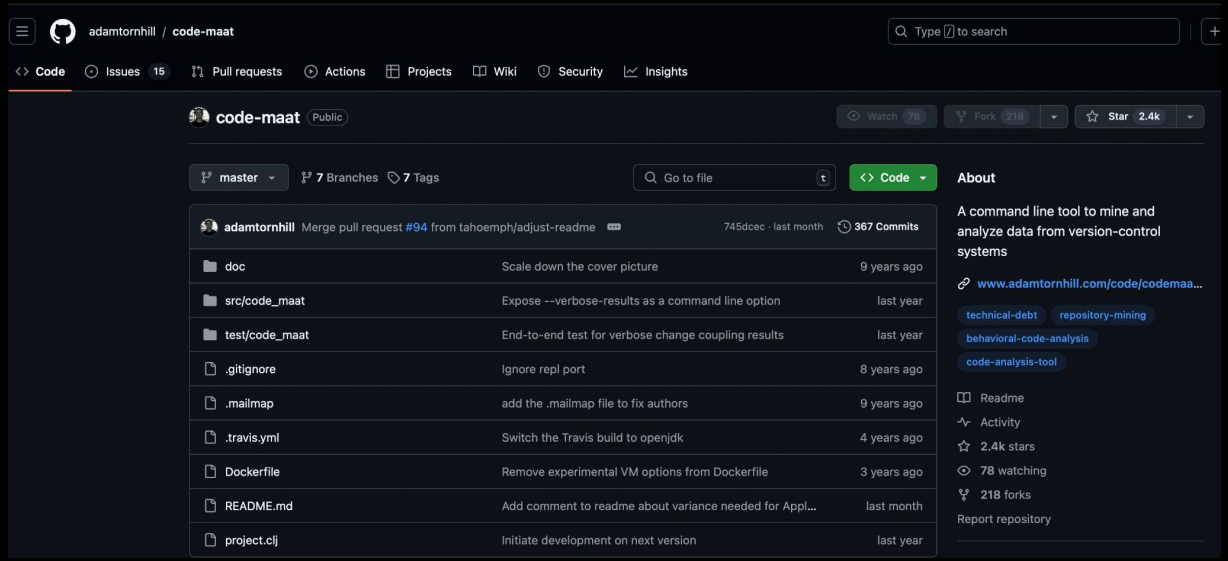
https://www.youtube.com/watch?v=90l0hH5-r5A

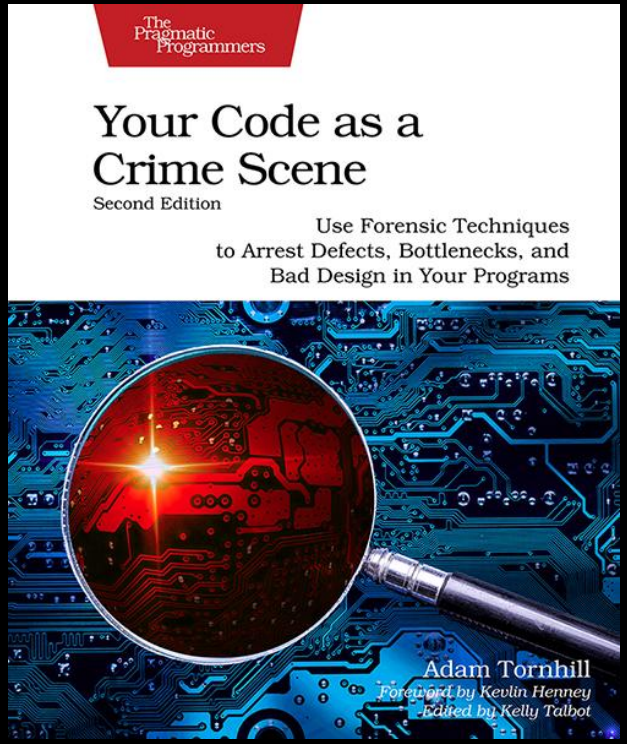
A Case-study in Rewriting a Legacy GUI Library for Real-time Audio Software in Modern C++

What is legacy code?

What is legacy code?

- No tests
- Lot’s of code
- Very old
- Authors may be gone
- Many C++ standards
- New/old styles
- New/old paradigms
- Bad decisions from the past that once made sense
- Possibly rewritten by a less skilled eng. org.





Your Code as a Crime Scene + `code-maat`

How big is the universe?

~15 Millions Lines of C/C++/Objective-C(++)

~670,000 Lines

Product Code

- Ozone
- RX
- Neutron
- Nectar
- etc.

~1.33 Million Lines

Shared Code

- iZBase
- iZDSPBase
- Glass
- EqualizerIIR
- etc.

~13 Million Lines

Open Source

- Boost
- Skia
- libPNG
- libXML2
- etc.

NATIVE INSTRUMENTS

iZOTOPE

Plugin Alliance

BRAINWORX

A cautionary tale...

...of linting, typechecking, and unit tests

Story of maintaining some Python tools

- Adopt new Poetry + pyproject.toml project templates
  - New linters
  - New type-checking
  - Code changes needed to adopt

- A goal to increase test coverage
  - “Refactoring” to make code more testable

“X% of your code will be rewritten in 5 years.”

Kubrick, Stanley. *2001: A Space Odyssey*. (1968)

Lang, Fritz. *Metropolis*. (1927)

Don’t change code without user/business value

Evolving styleguides...

...with `clang-format`

Improving with static analysis

- Always provide a clang-format file
- git-hooks to automatically apply formatting to changes
- Formatting verified in CI

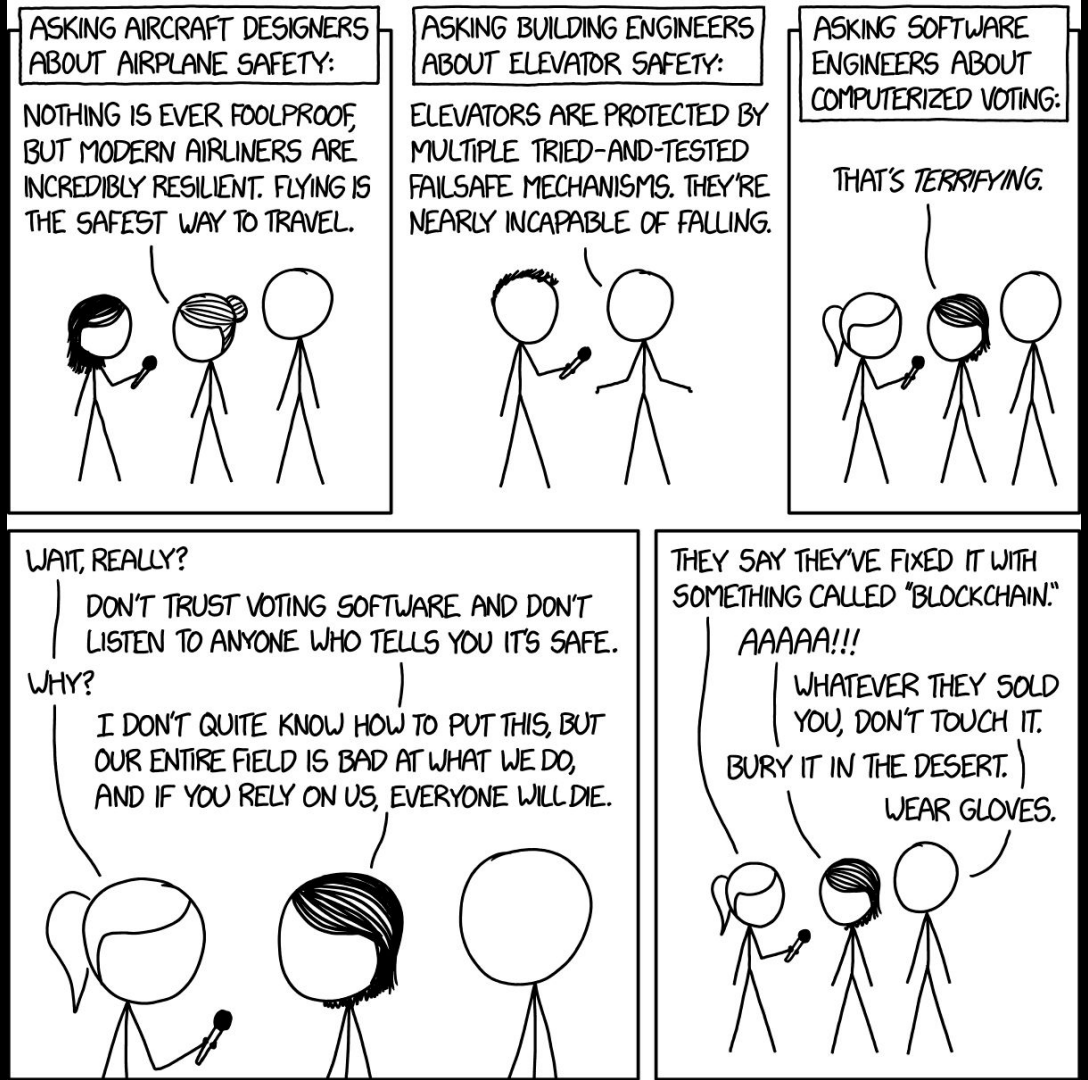
Improving with static analysis

...with ASAN, UBSAN, and TSAN


```
unsigned random() {
  unsigned x
  return x;
}
```

https://www.youtube.com/watch?v=vEtGtphl3Ic

Purging Undefined Behavior & Intel Assumptions in a Legacy C++ Codebase



xkcd.com/2030/

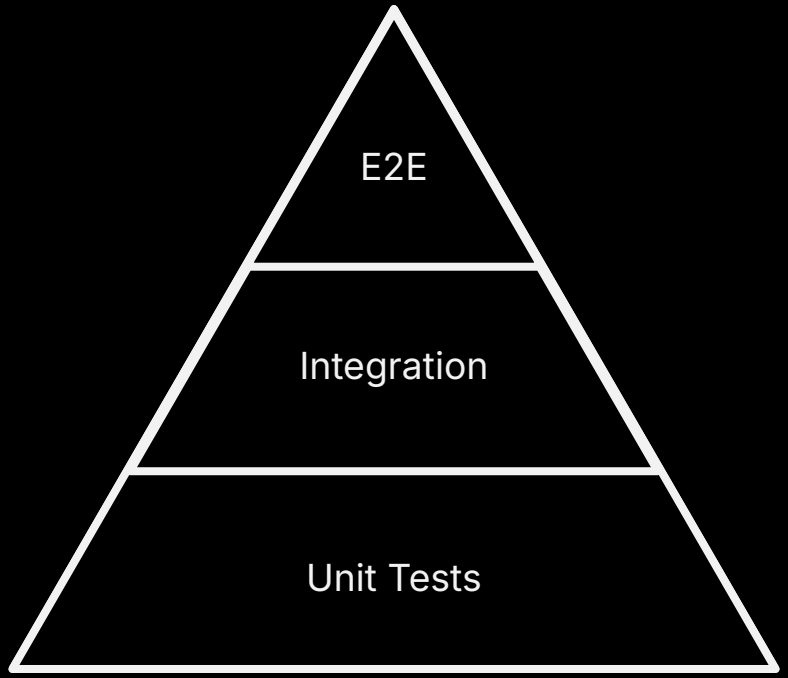
Improving with static analysis

Success
- Incremental rollout of ASAN / UBSAN

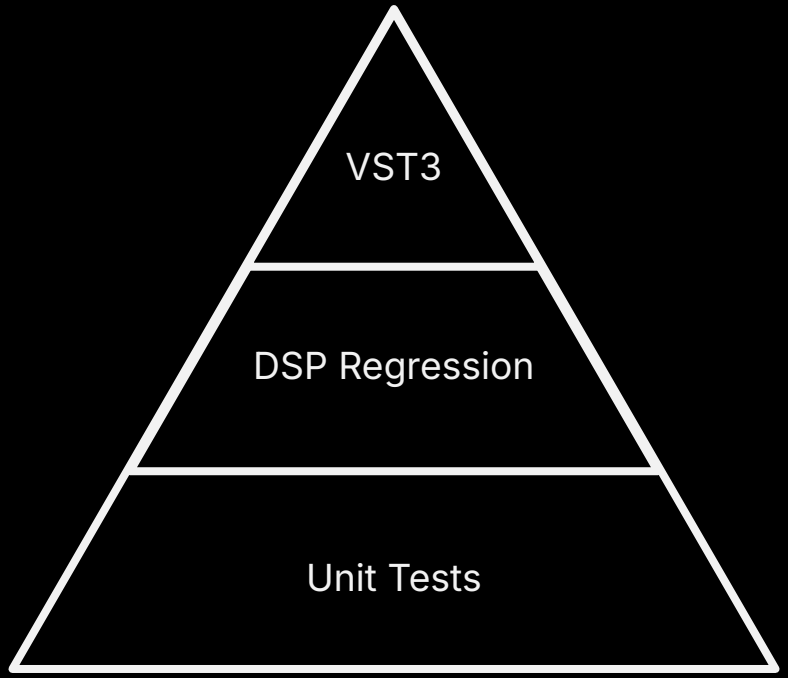
Failure?
- Full product testing with TSAN

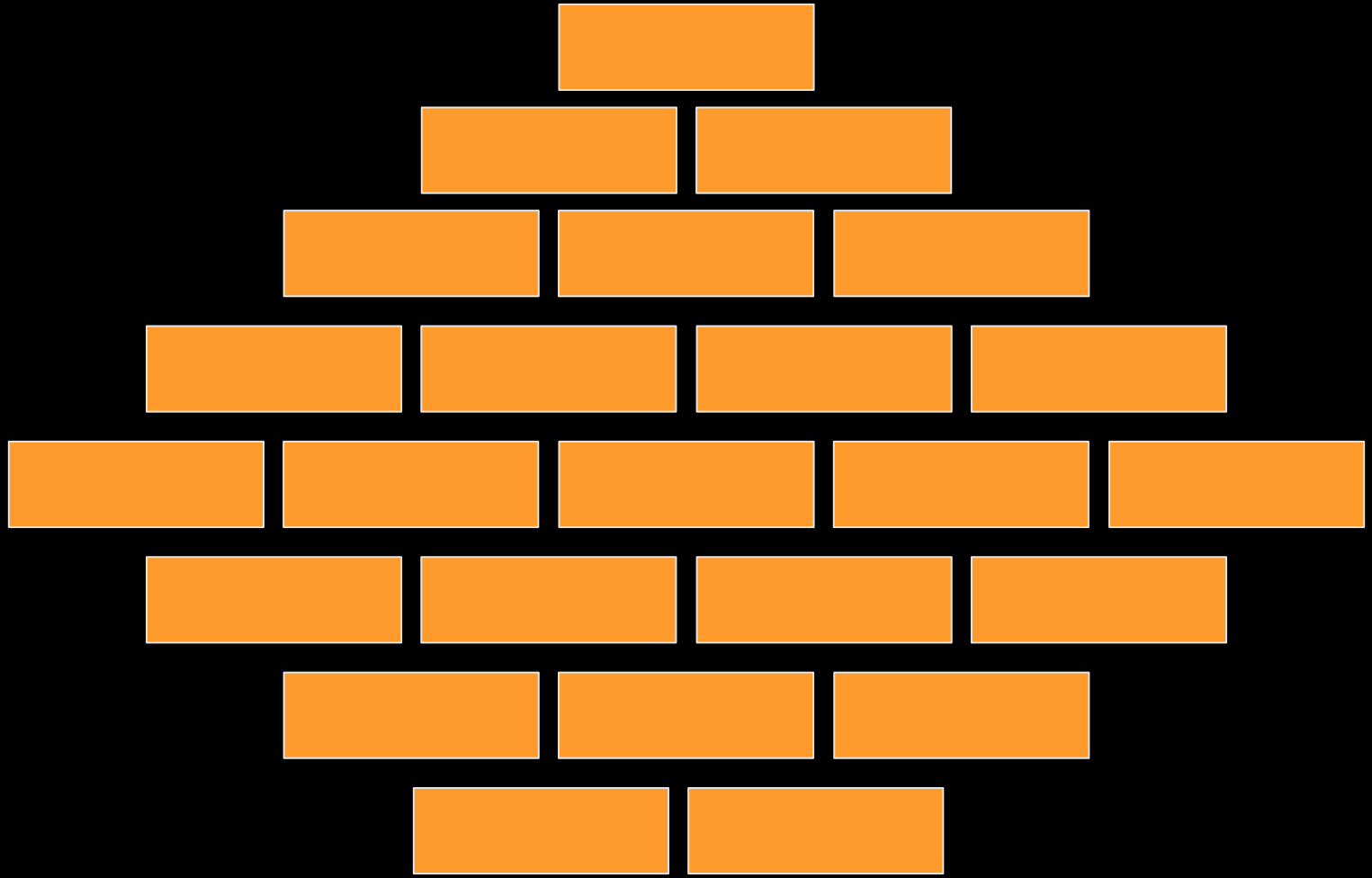
Write tests!

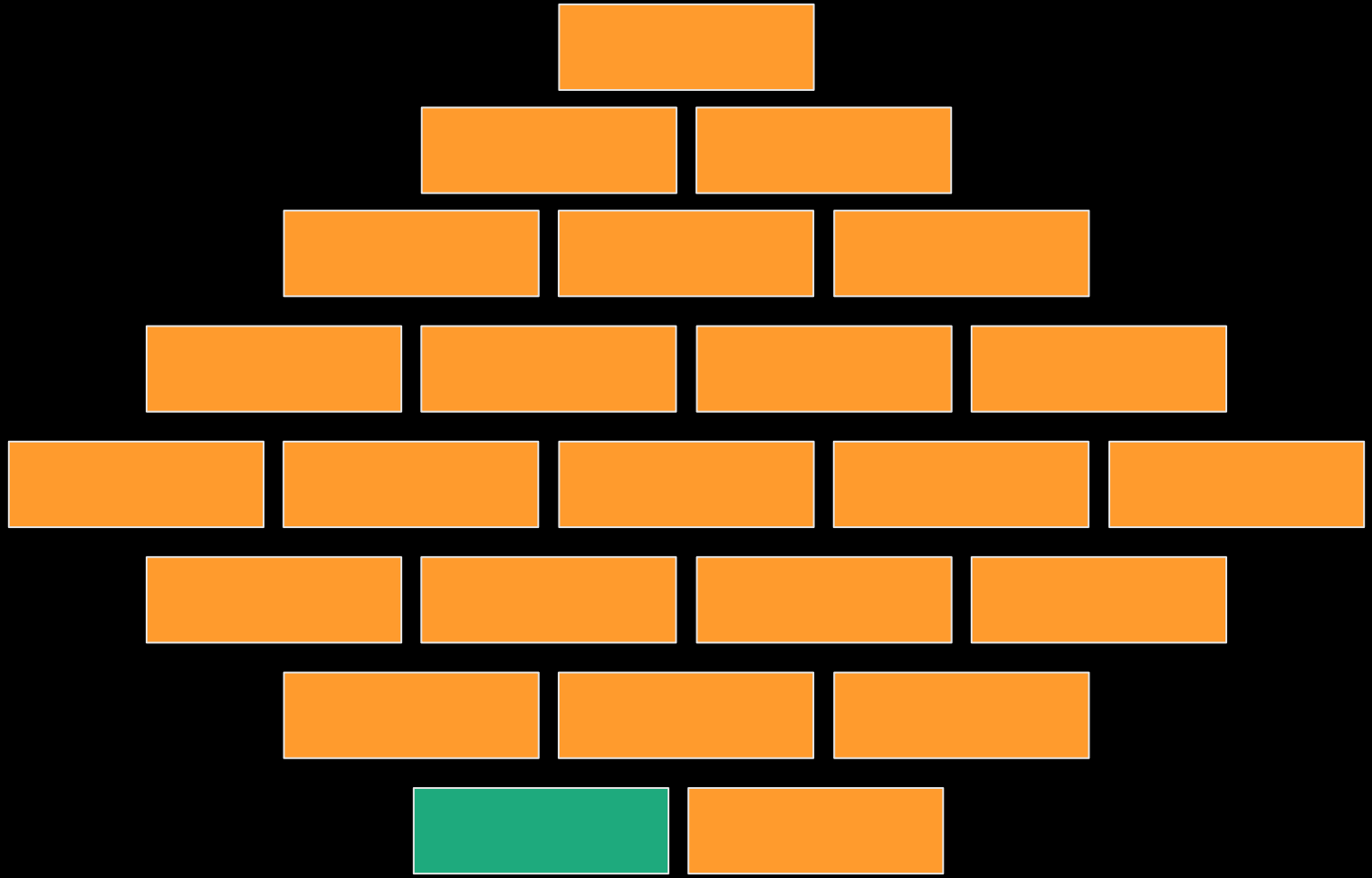
Moving up the testing pyramid

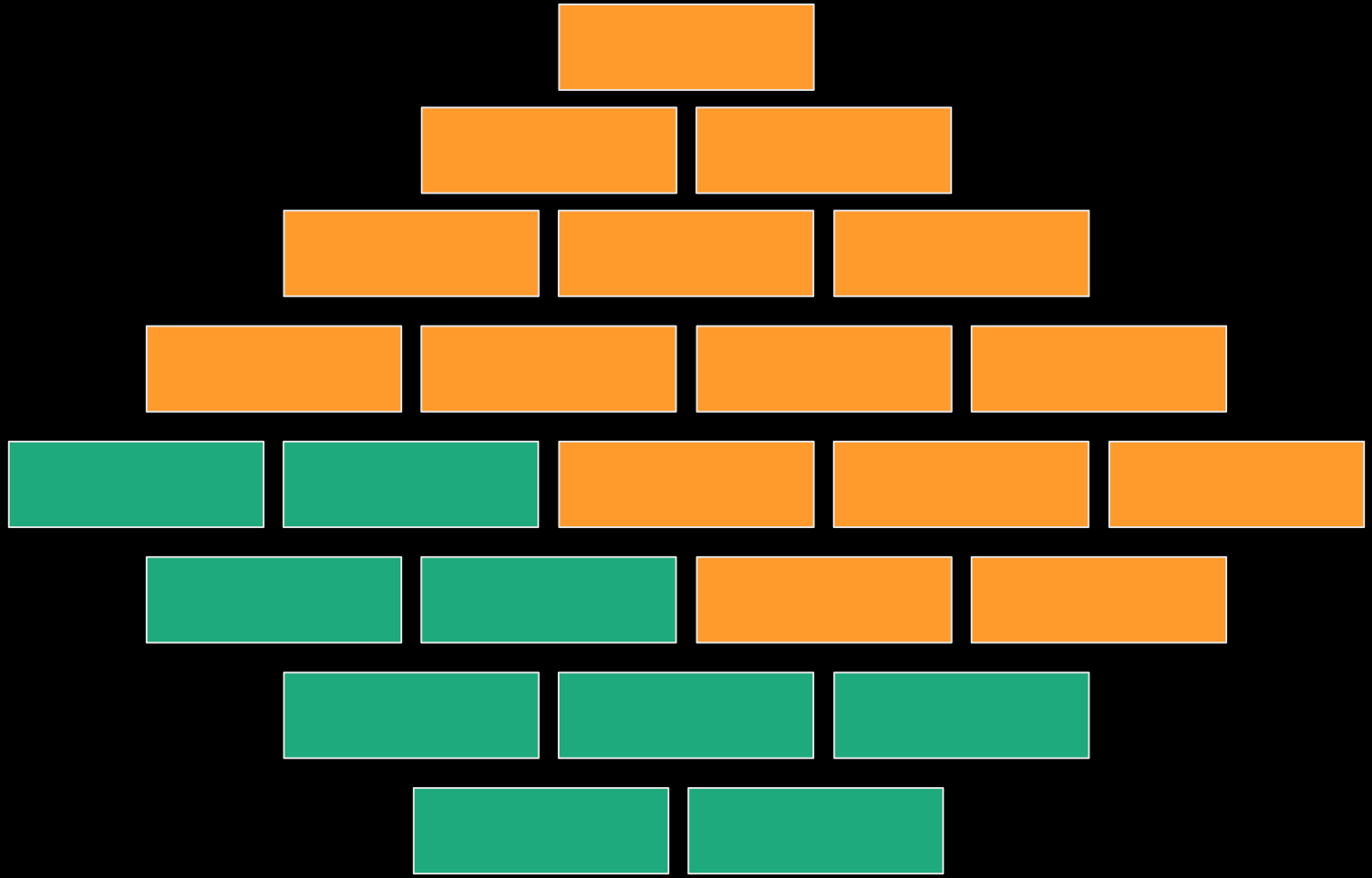


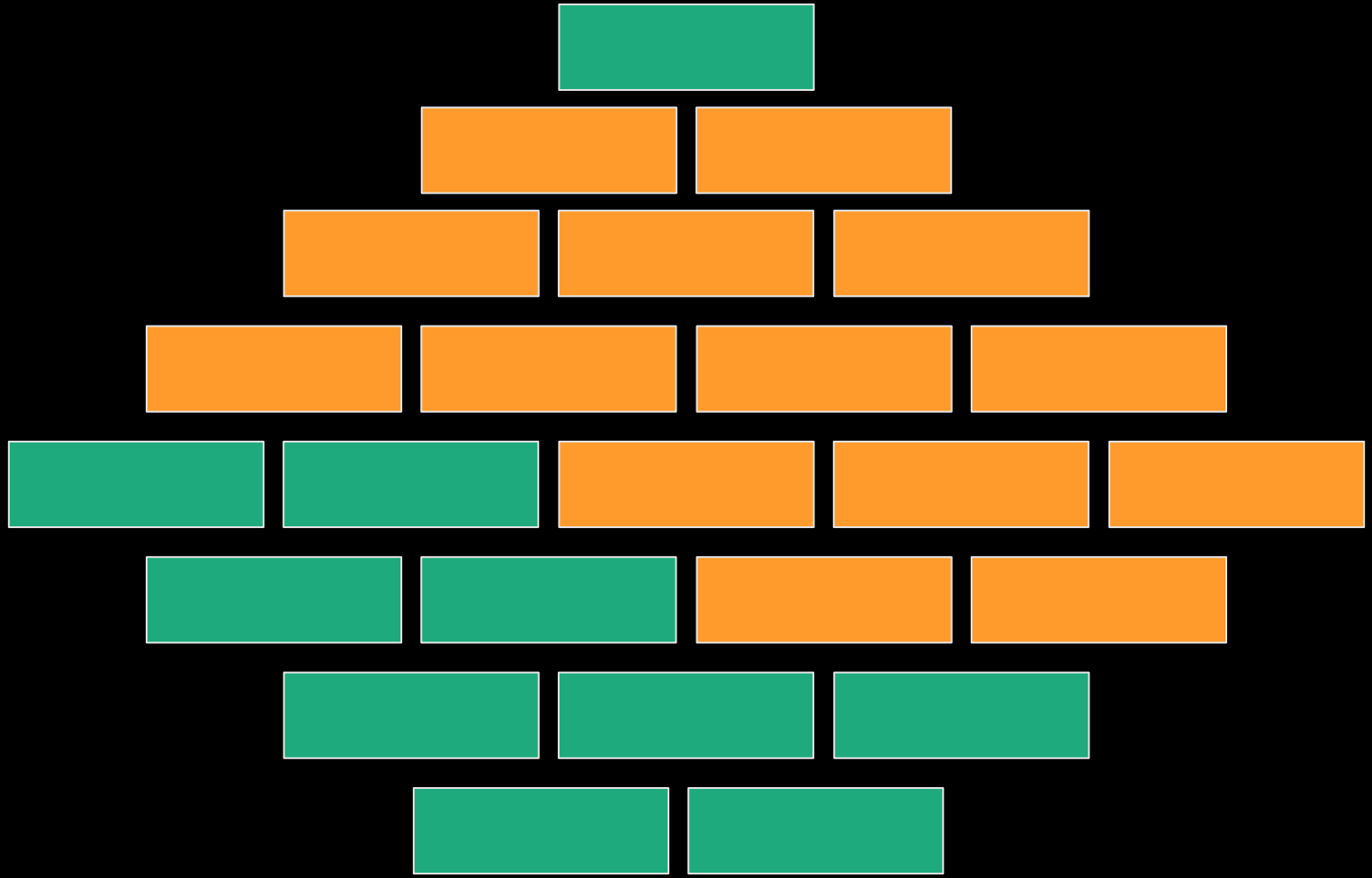
Moving up the testing pyramid











Changing legacy APIs

should you?

Breaking changes

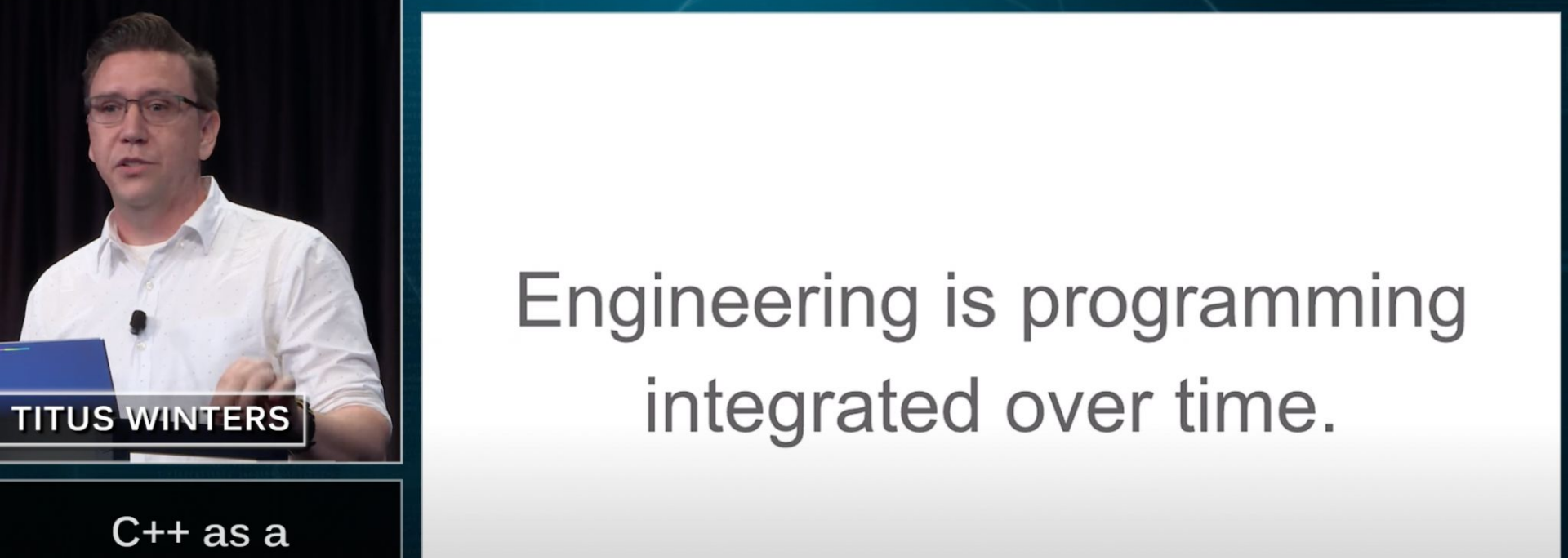
- class template <class T> checked_value
- class template <class T> boost::optional
- Platform transition:
  - using Optional = std::experimental::optional
  - using Optional = boost::optional
- std::optional

xkcd.com/1172/

No breaking changes

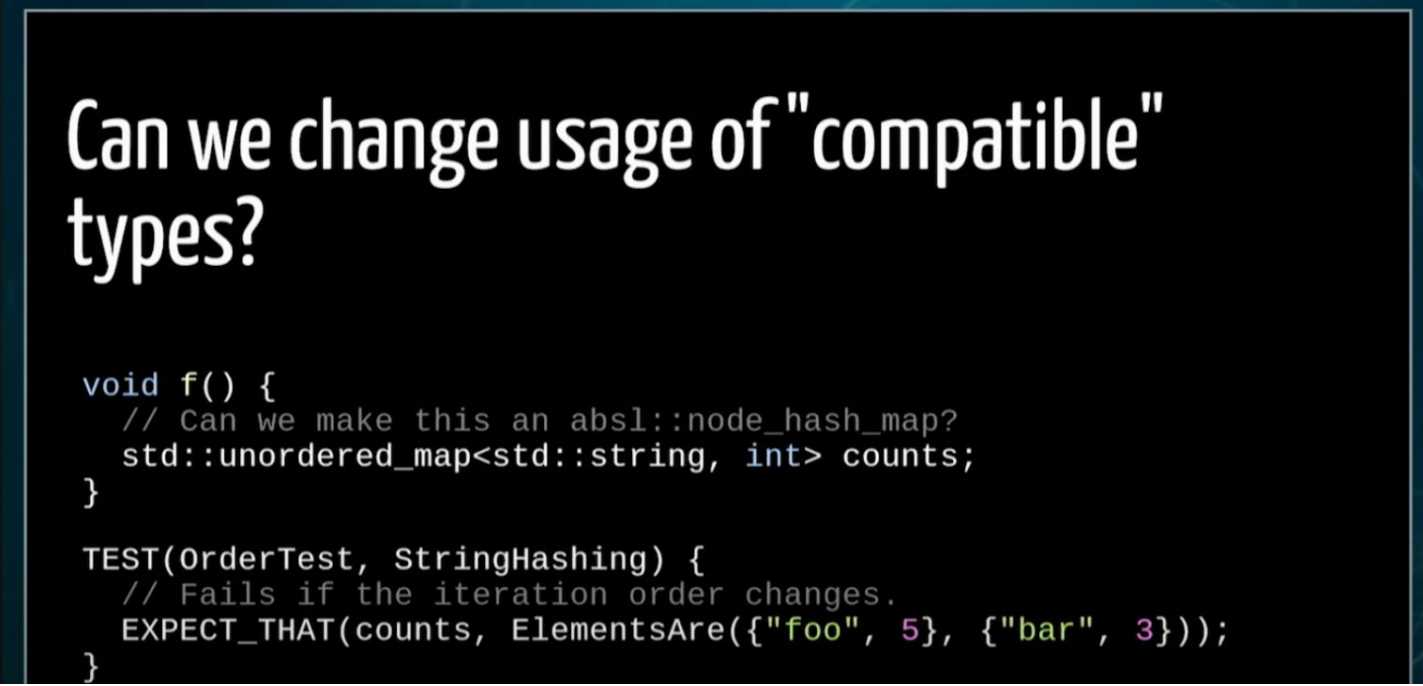
No breaking changes
(or use small libraries)

No breaking changes
(or provide transition plan)



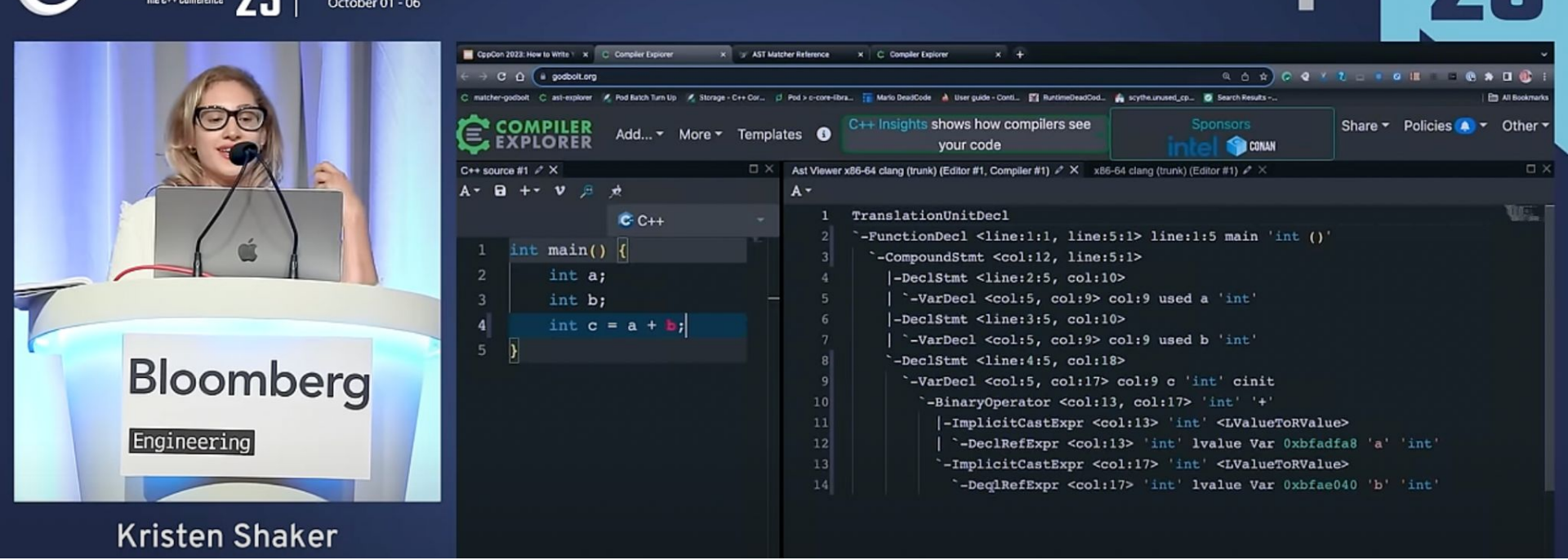
https://www.youtube.com/watch?v=tISy7EJQPzI

C++ as a "Live at Head" Language



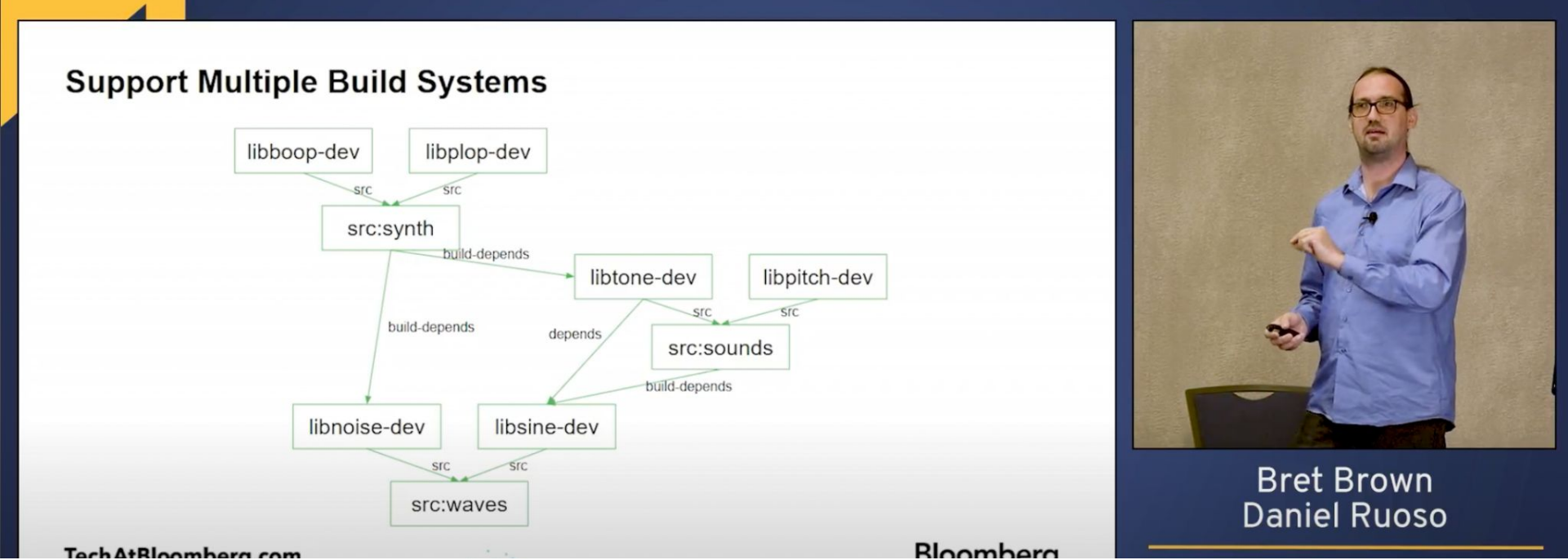
https://www.youtube.com/watch?v=v_yzLe-wnfk

Maintainability and Refactoring Impact of Higher-Level Design Features



https://www.youtube.com/watch?v=torqlZnu9Ag

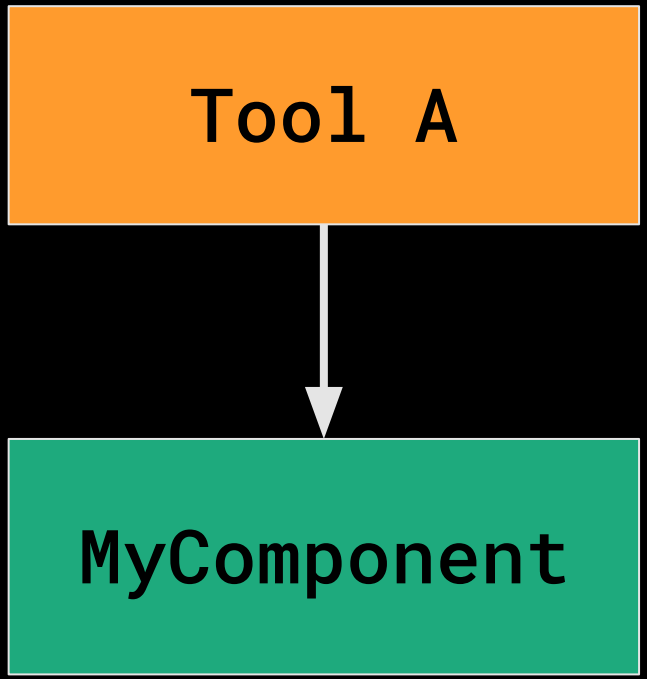
How to Build Your First C++ Automated Refactoring Tool



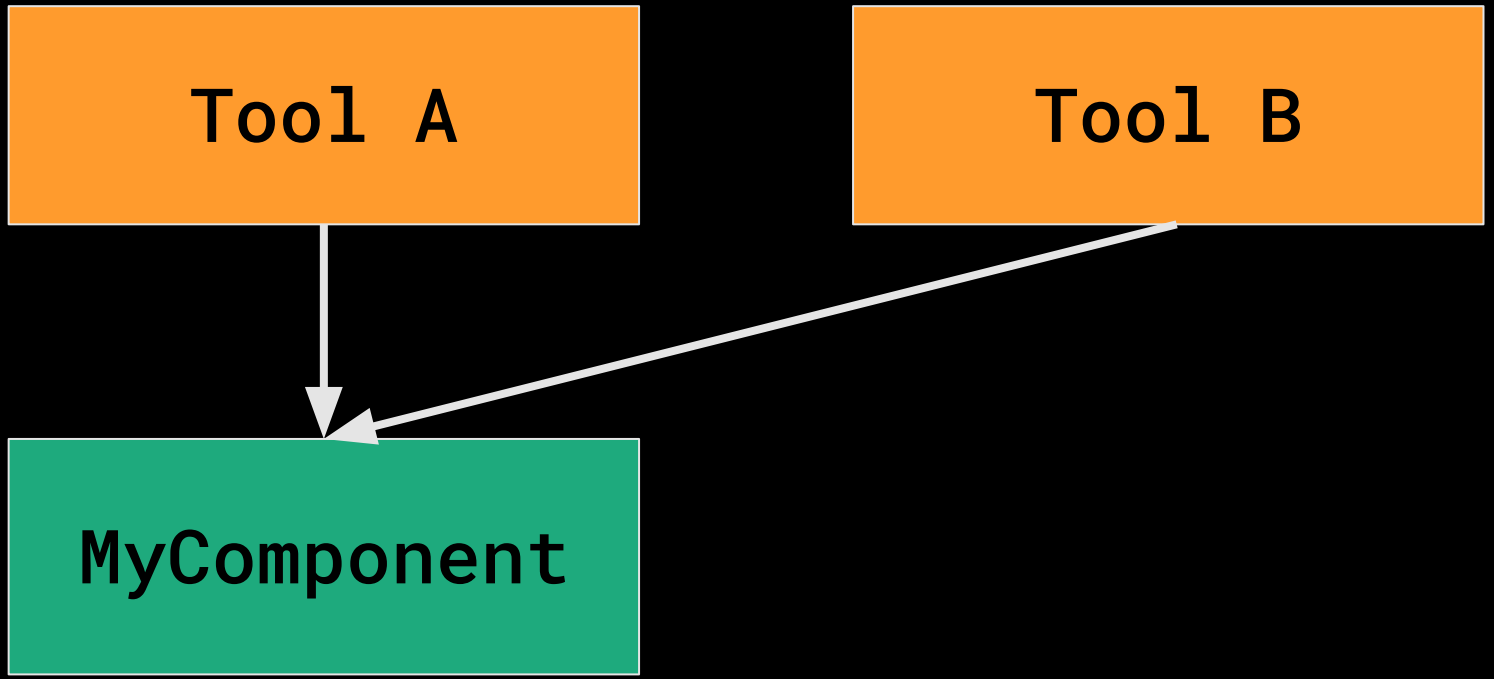
https://www.youtube.com/watch?v=R1E1tmeqxBY

Lessons Learned from Packaging 10,000+ C++ Projects

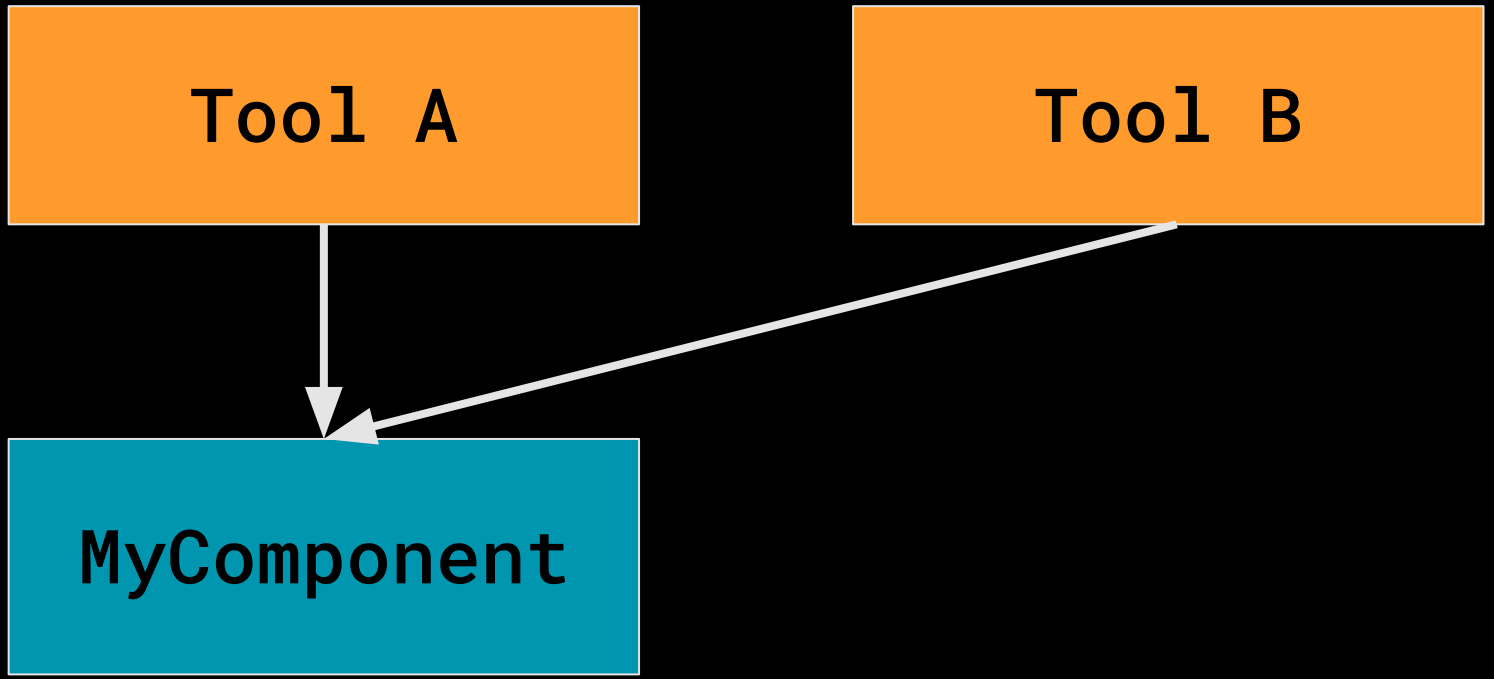
Idea from Lakos, John. *Large Scale C++: Process and Architecture, Volume 1*(2020).



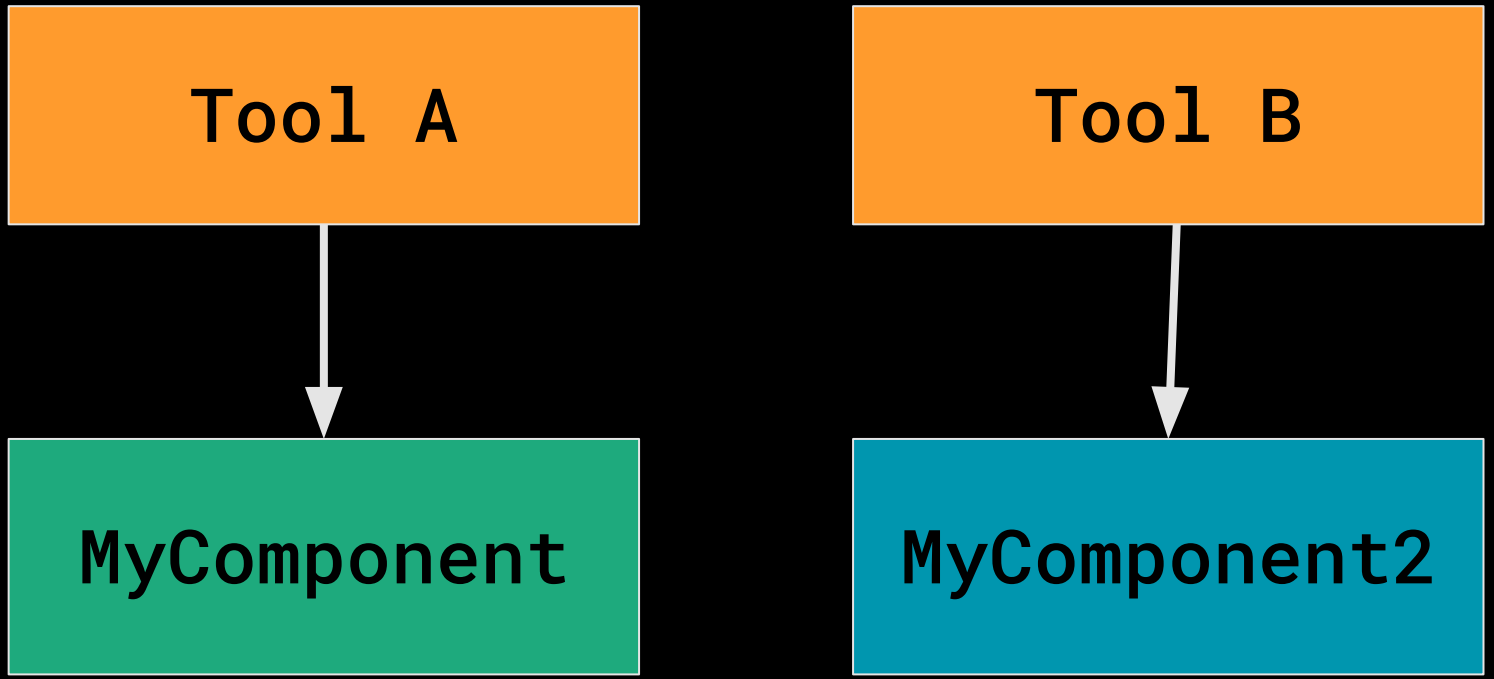
Idea from Lakos, John. *Large Scale C++: Process and Architecture, Volume 1*(2020).



Idea from Lakos, John. *Large Scale C++: Process and Architecture, Volume 1*(2020).



Idea from Lakos, John. *Large Scale C++: Process and Architecture, Volume 1*(2020).



Static analysis and handling deprecations

...with `clang-tidy`

Improving with runtime analysis

- clang-tidy-diff lets you run only on changed lines
- clang-compilation database may need modifications
  - replace PCH with real header
  - compile header as Objective-C++
- Turn on deprecation warnings as errors
  - can be used to incrementally adopt warnings
- verify in CI
- have a way to disable this check in CI to deprecate

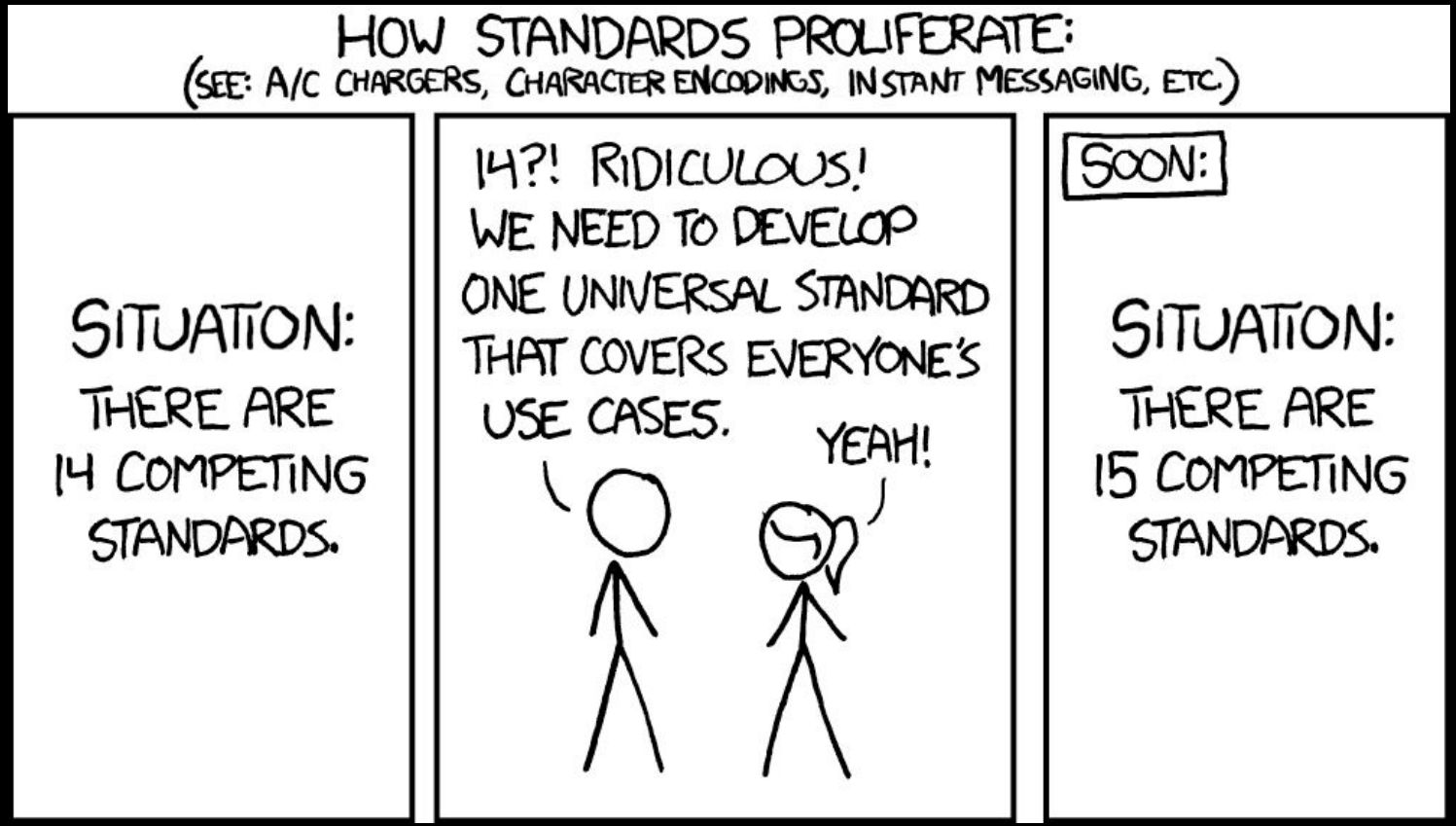
#red-diff-appreciation-society

Sharing Legacy Code

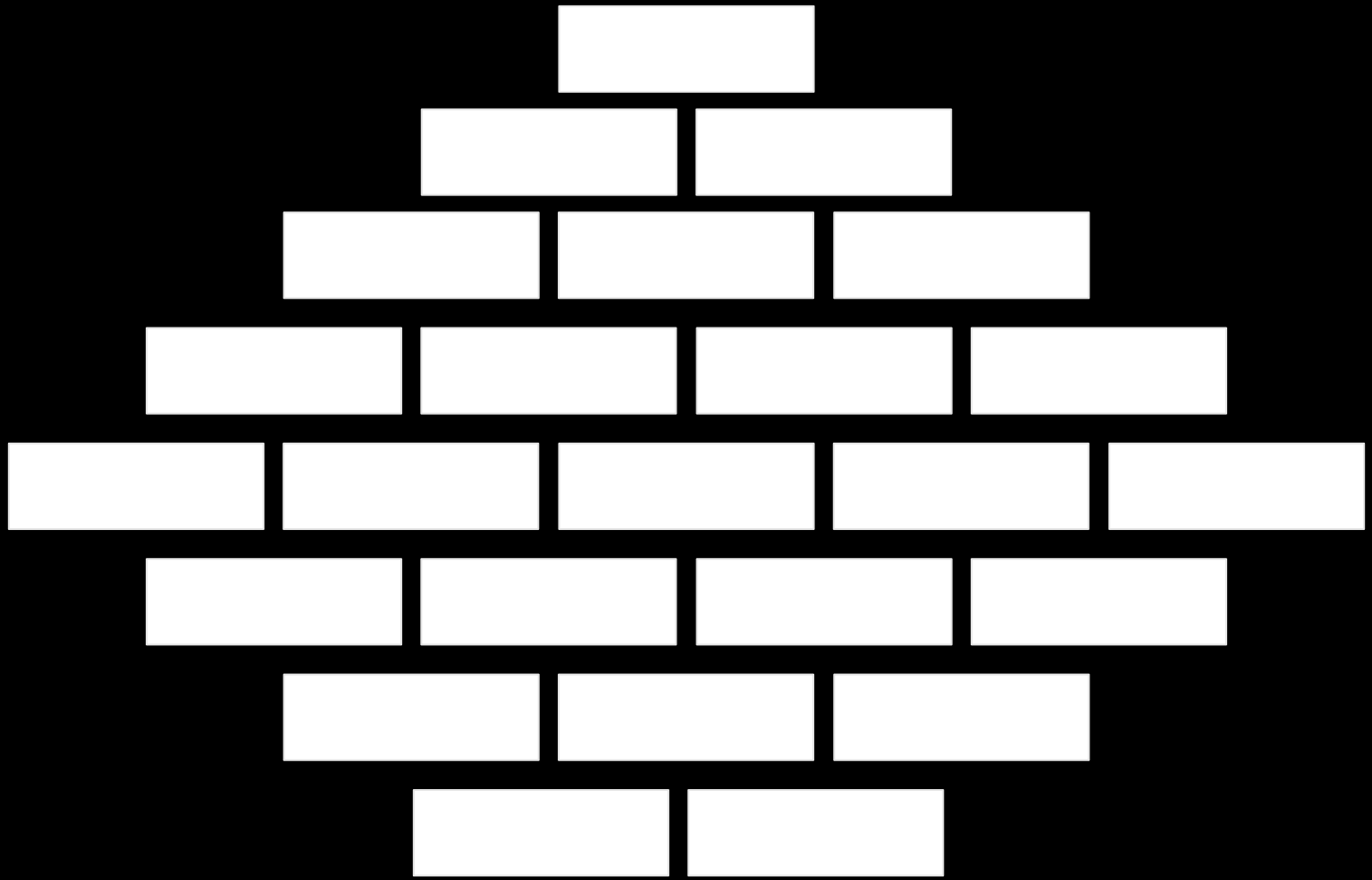
The desire to standardize

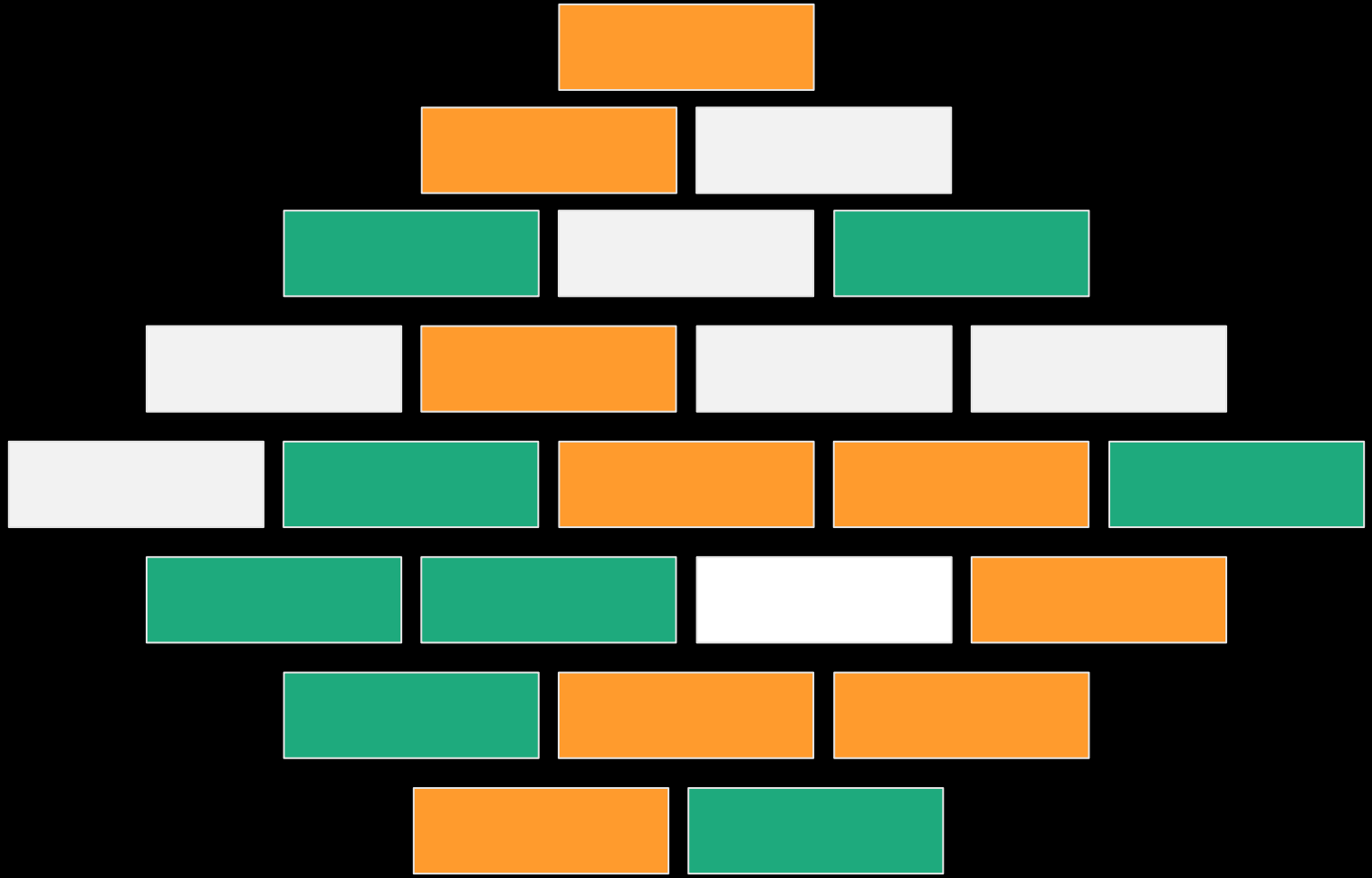
We have built the same things...

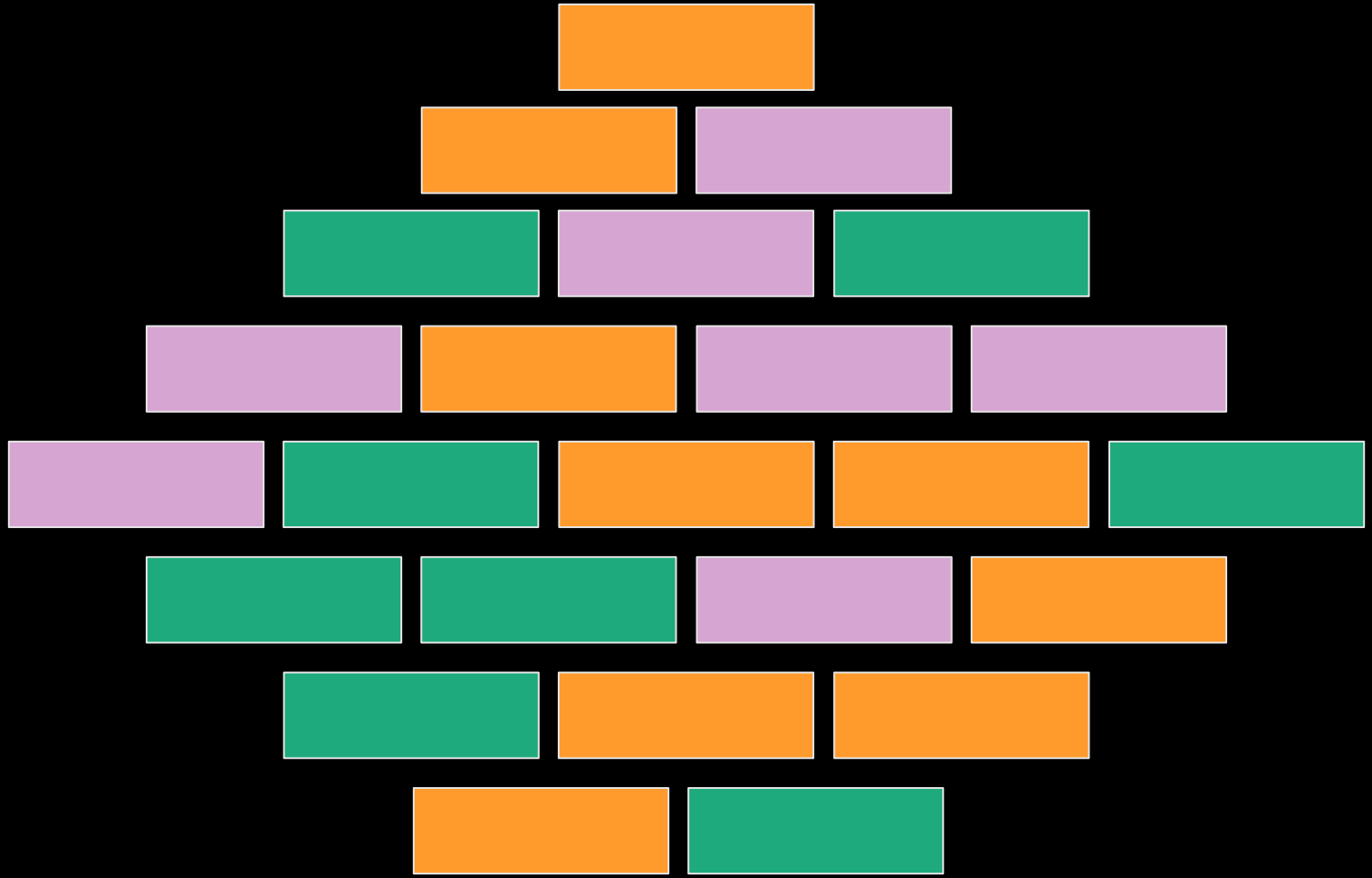
- Lots of similar and unique DSP algorithms
- 7 UI technologies
- 4 installer technologies
- etc....



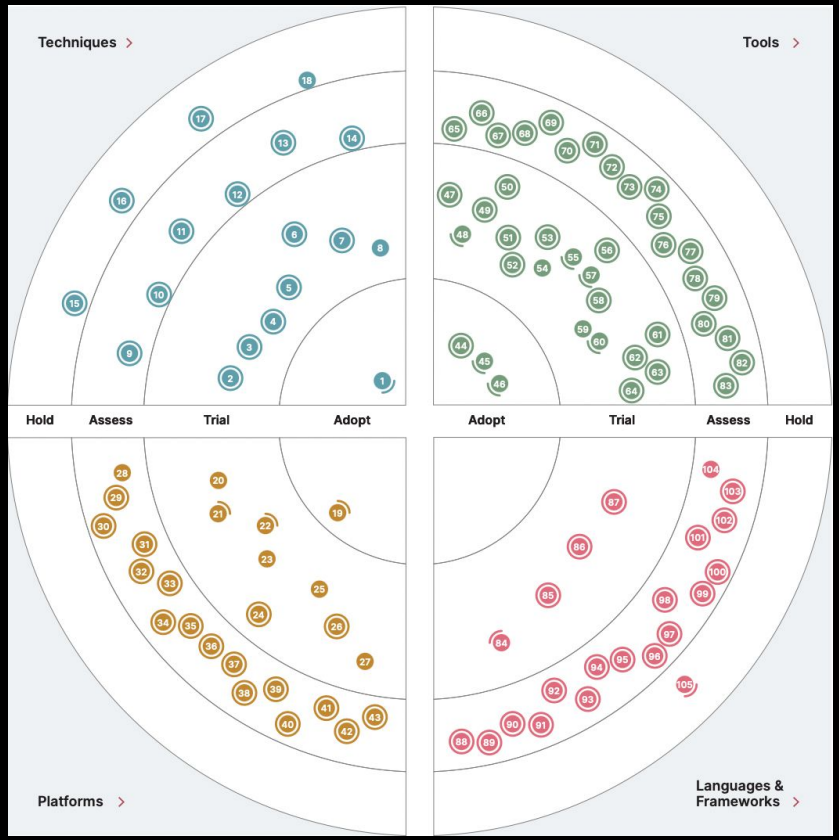
xkcd.com/2030/







Discover your hidden gems



https://www.thoughtworks.com/en-us/radar

What challenges come with sharing legacy code?

What challenges come with sharing legacy code?

- Different build systems
- Possibly different languages standards
- Different versions of 3rd-party libraries
- Precompiled headers
- Weird legacy vocabulary types
- Global namespace solutions
- Different error handling strategies
- Different code styles

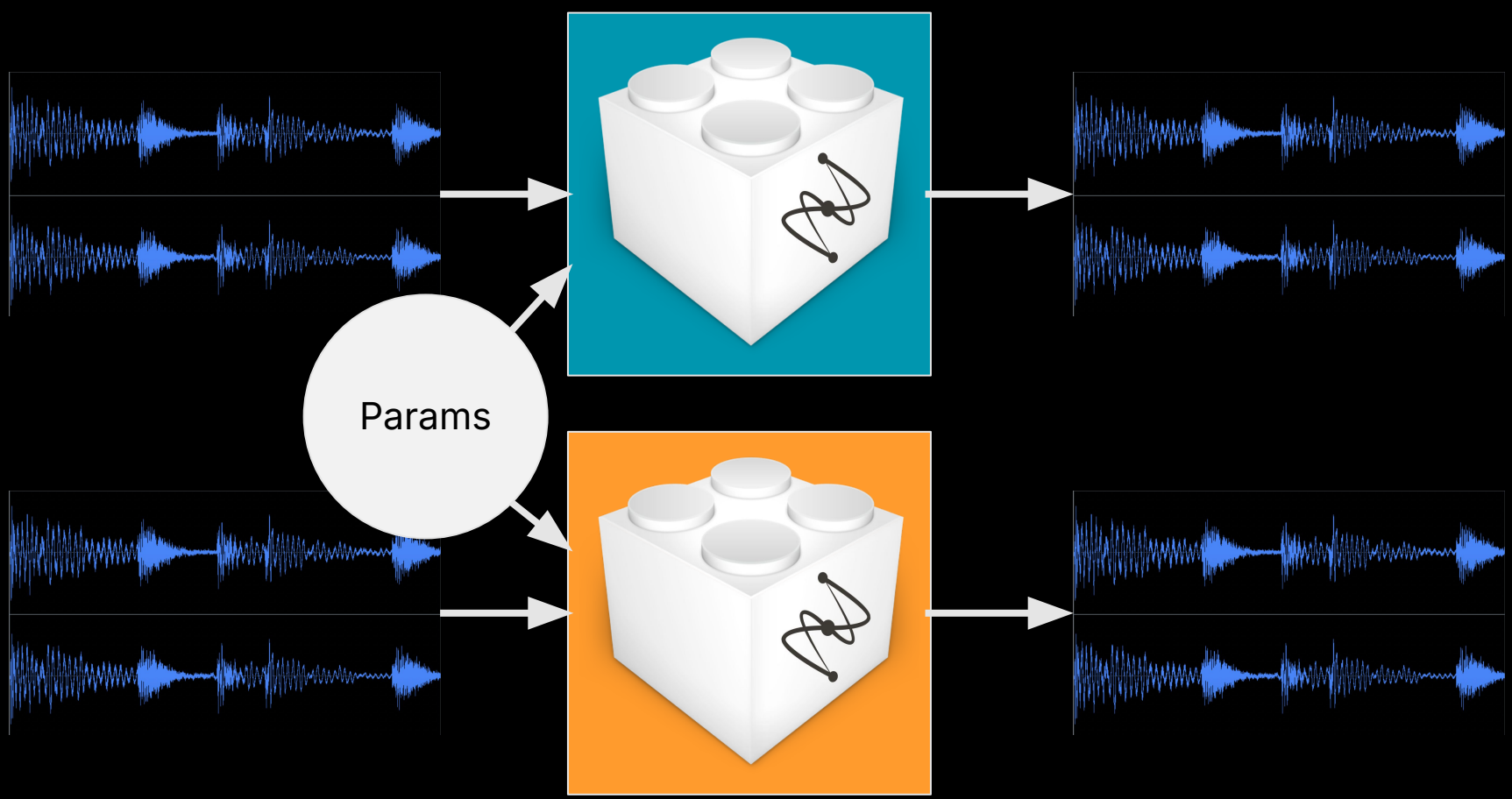
Please provide a
`.clang-format`

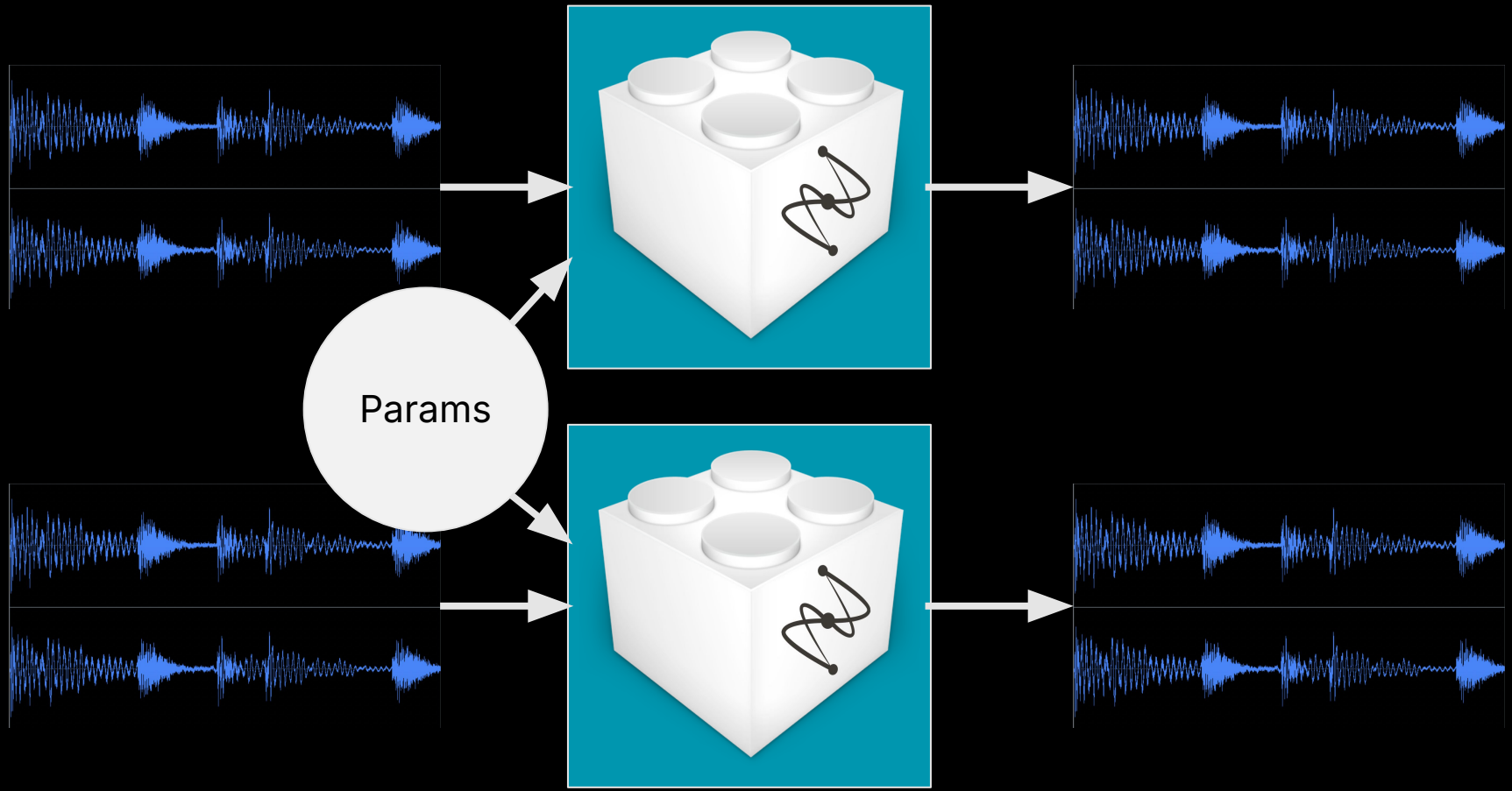
Dynamic library sharing

...with TestableDSP (TDSP)


Ozone
ADVANCED
Presets
Delta
Learn Input Gain
-11.0 LUFS
IRC 1
+12.0 dB
Gain
Output Level
0.00
True Peak
Character
Fast and Loud
2.00
-1.9
Upward Compress
0.0 dB
Soft Clip
0%
H
M
L
Transient Emphasis
0%
Stereo Independence
Transient
0%
Sustain
0%
-4.3 -3.8 Peak 0.0 0.0
-12.8 -13.9 RMS -1.2 -2.3
Bypass
Dither


TRAKTOR PRO 4
In Dub
Cooking Swings
Internal Logic
Gabriel Montufar featuring Nicole Torres - Fade Away (Cory G
Track Collection
Playlists
Beatport Streaming
Beatsource Streaming
Explorer
Audio Recordings
Music
History
Bumpy
A Wee Dub
Bae
Diafonie
Treatwave
Words
Born Under
Internal Logic
Cooking Swings


Params


Params

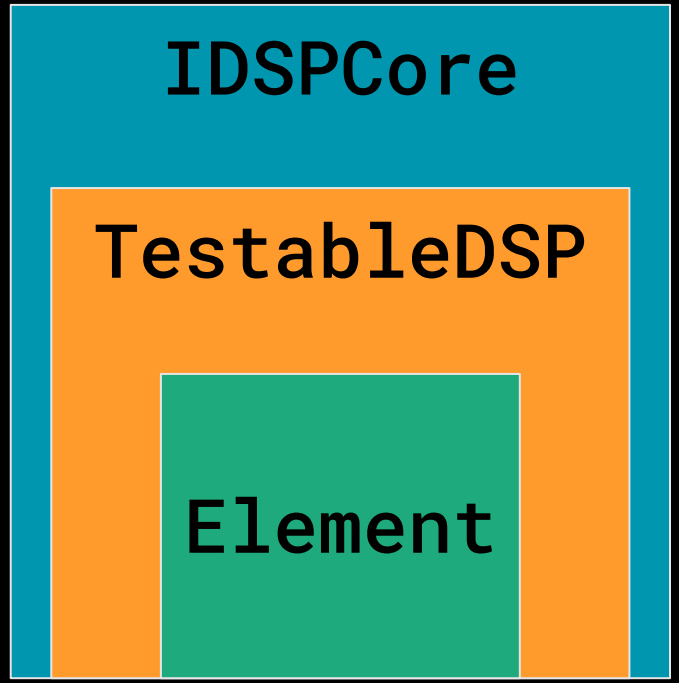
TestableDSP Interface

```
TestableDSP* TestableDSP_Create(
                unsigned instance,
                unsigned cargs,
                const char* vargs[]);

void TestableDSP_Destroy(TestableDSP* tdsp);
```

TestableDSP Interface

```
class TestableDSP {
public:
    virtual void SetBool(size_t index, bool value) = 0;
    virtual void SetFloat(size_t index, float value) = 0;
    virtual void SetEnum(size_t index, unsigned value) = 0;

    virtual unsigned GetNumChannels() = 0;

    virtual float GetSamplingRate() = 0;

    virtual void Reset() = 0;

    virtual void Process(size_t n, float* const* buf) = 0;
};
```

A successful pattern!



TestableDSP Interface

```
class __attribute__((visibility("default"))) TestableDSP {
public:
    virtual void SetBool(size_t index, bool value) = 0;
    virtual void SetFloat(size_t index, float value) = 0;
    virtual void SetEnum(size_t index, unsigned value) = 0;

    virtual unsigned GetNumChannels() = 0;

    virtual float GetSamplingRate() = 0;

    virtual void Reset() = 0;

    virtual void Process(size_t n, float* const* buf) = 0;
};
```

TestableDSP Interface

```
class __declspec(dllexport) TestableDSP {
public:
    virtual void SetBool(size_t index, bool value) = 0;
    virtual void SetFloat(size_t index, float value) = 0;
    virtual void SetEnum(size_t index, unsigned value) = 0;

    virtual unsigned GetNumChannels() = 0;

    virtual float GetSamplingRate() = 0;

    virtual void Reset() = 0;

    virtual void Process(size_t n, float* const* buf) = 0;
};
```

Source code sharing

Two stories of source code sharing

- Simple ML guitar amp modeling (not production)
  - Can be converted to header only
  - Depends on eigen (we already use)

- Adopting Native Instruments' auth library
  - Split out from monolithic nilibs library
  - Replace custom vocab types with std::
  - Align on 3rd-party libraries/versions
    - Boost, base64, nlohmann/json
  - Build scripts for both build systems

Copy+Paste sharing


STAGE 1
STAGE 2
SHAPE
FILTER
Zoom
Type
TEN INCH SPIKE
Gnasher
Rubber Hammer
Sandpaper
Slight Twist
Soft Taco
Sputter
Squealer
Tarnish
Ten Inch Spike
Triangle Clipper
Weathered
Clear All
Remove
Linear
Bipolar
Log mode
DC Filter
Base
Your curve
Result
Multiband
Pre
Drive
Style
Gain
Mix
6.9
2.3
0.50
0.3
100.0
Solo
Bypass
Options
History
Graph
Reset
INPUT
OUTPUT
LIMITER
MIX
DRY
WET
TRASH2
FILTER 1
TRASH
FILTER 2
CONVOLVE
DYNAMICS
DELAY
iZotope
PRESETS
BYPASS


kOLor
Diode
Drive
Gain Boost
Bass Saver
Bass
Mid
Treble
LP
HP
Mix
Volume

Static library sharing

...sharing an analytics library

- Reactive programming event dispatch

- Event dispatch
- local caching
- opt-in management
- user-id management

Static library sharing

Source code sharing blockers

- Util::ConfigFile (plist/registry abstraction)
  - String

Dynamic library sharing

- We know it works
- Decided on layer for abstract API for virtual interface
- Would require installer tech investment to deliver
- Long term goal is source code sharing

Static library sharing gotchas

- Dummy dynamic library to generate static libs
- No LTO

macOS

- -fno-objc-msgsend-selector-stubs
- -Qunused-arguments

Windows

- Agree on iterator debug level
- Static or Dynamic RT
- Enable MultiThreadDLL
- Other flag dances

Summary

Incremental improvement

- Focus on improving new code; don’t fix everything
- Incrementally adopt clang-format (clang-format-diff)
- Incrementally adopt sanitizers
- Incrementally adopt clang-tidy (clang-tidy-diff)

Bonus content...

...how this talk inspired me (source sharing)

Static library sharing

Source code sharing blockers

- Util::ConfigFile (plist/registry abstraction)
  - String

Dynamic library sharing

- We know it works
- Decided on layer for abstract API for virtual interface
- Would require installer tech investment to deliver
- Long term goal is source code sharing

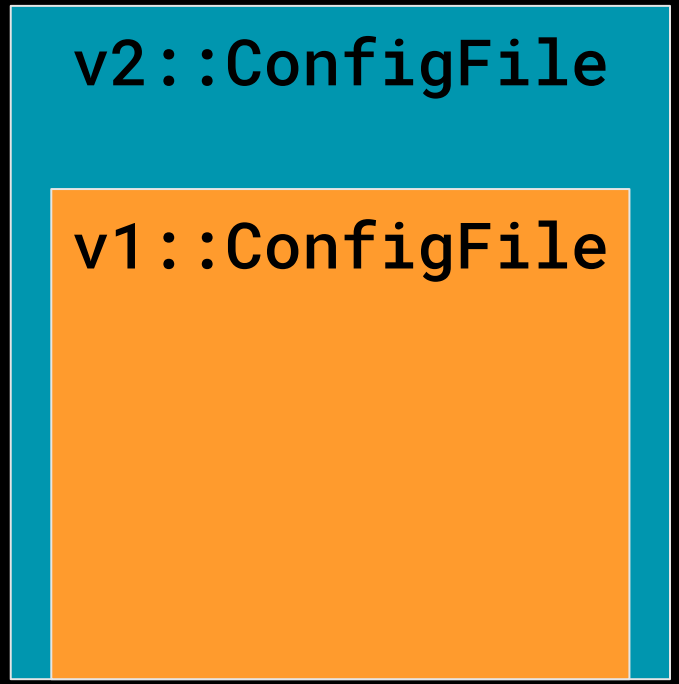
```
namespace v1 {
class ConfigFile {
public:
    explicit ConfigFile(const String& root);

    virtual String GetString(const String& key,
                             const String& default);

    virtual bool SetString(const String& key,
                           const String& value);

    // more getters/setters

    virtual bool Synchronize();
};
}
```

```
namespace v2 {
class ConfigFile {
public:
    explicit ConfigFile(std::string_view root);

    virtual std::string GetString(std::string_view key,
                                  std::string_view default);

    virtual bool SetString(std::string_view key,
                           std::string_view value);

    // more getters/setters

    virtual bool Synchronize();
};
}
```

```
bool readString(const String& appID_, const String& value, String& str ) {
    CFString appID{appID_};

    CFString cfstrKey{normalizeCFPrefName(value)};
    CFTypeRef cftype = CFPreferencesCopyAppValue(cfstrKey, appID);

    if (!cftype) {
        str = String{};
        return false;
    }
    Sentry::ReleaseCFRef s_cftype{cftype};

    if (CFGetTypeID(cftype) != CFStringGetTypeID()) {
        str = String{};
        return false;
    }
    CFStringRef cfstr_ref = reinterpret_cast<CFStringRef>(cftype);

    CFString cfstr{cfstr_ref, CFString::no_release};
    str = cfstr.ToString();
    return true;
}
```

That’s one function!

A successful pattern!



```
namespace v1 {
class ConfigFile {
public:
    explicit ConfigFile(const String& root);

    virtual String GetString(const String& key,
                             const String& default);

    virtual bool SetString(const String& key,
                           const String& value);

    // more getters/setters

    virtual bool Synchronize();
};
}
```

```
namespace v2 {
class ConfigFile {
public:
    explicit ConfigFile(std::string_view root);
    explicit ConfigFile(v1::ConfigFile&& cf)
        : m_cf{std::move{cf}} {}

    virtual std::string GetString(std::string_view key,
                                  std::string_view default) {
        return m_cf.GetString(String{key}, String{default});
    }
    virtual bool SetString(std::string_view key,
                           std::string_view value) {
        return m_cf.SetString(String{key}, String{value});
    }
    virtual bool Synchronize() { return m_cf.Synchronize(); }
};
}
```

Source sharing unblocked

Thank you!

Roth Michaels
Principal Software Engineer
roth.michaels@native-instruments.com
@thevibesman