

C++ Build Systems

Thomas Deniffel

Who of you is
a C++-
Programmer?





Who is using CMake?

Using Cmake with Make?



Using CMake with Ninja?





Other CMake?



Bazel / Buck?



C++ Build Systems

Thomas Deniffel



Build
Systems are
hard

<outline>

- * Basics
- * Concepts
- * Comparing

**5 categories
sub-elements**

`github.com/
tom-010/
cpp_build_systems/`

Thomas Deniffel

- Programmer
- Skytala
- Drones
- Meetups





**Who is a
developer?**

**Who came to
good food?**



```
g++ -o calculator \
calculator.cpp \
add.cpp sub.cpp \
mul.cpp div.cpp
```

```
echo \  
“g++ -o calculator \  
calculator.cpp \  
add.cpp sub.cpp \  
mul.cpp div.cpp” \  
> build.sh && \  
chmod +x build.sh
```

```
$CXX -o calculator \
calculator.cpp \
add.cpp sub.cpp \
mul.cpp div.cpp
```

```
$cat Makefile
```

```
calculator: calculator.cpp add.cpp  
cc -o calculator calculator.cpp add.cpp
```

```
$ make -j8
```

```
SOURCES = calculator.cpp add.cpp

OBJECTS = $(SOURCES:.c=.o)

calculator: $(OBJECTS)
           $(CXX) -o $@ $^
           -include $(SOURCES:.c=.d)

%.d: %.c
      @$(CC) -MM $(CPPFLAGS) &< | \
sed 's#\(\.*\)\.o: #1.o \1.d: #g' > $@
```

Makefile:6: * missing separator
(did you mean TAB instead of 8
spaces?). Stop.**

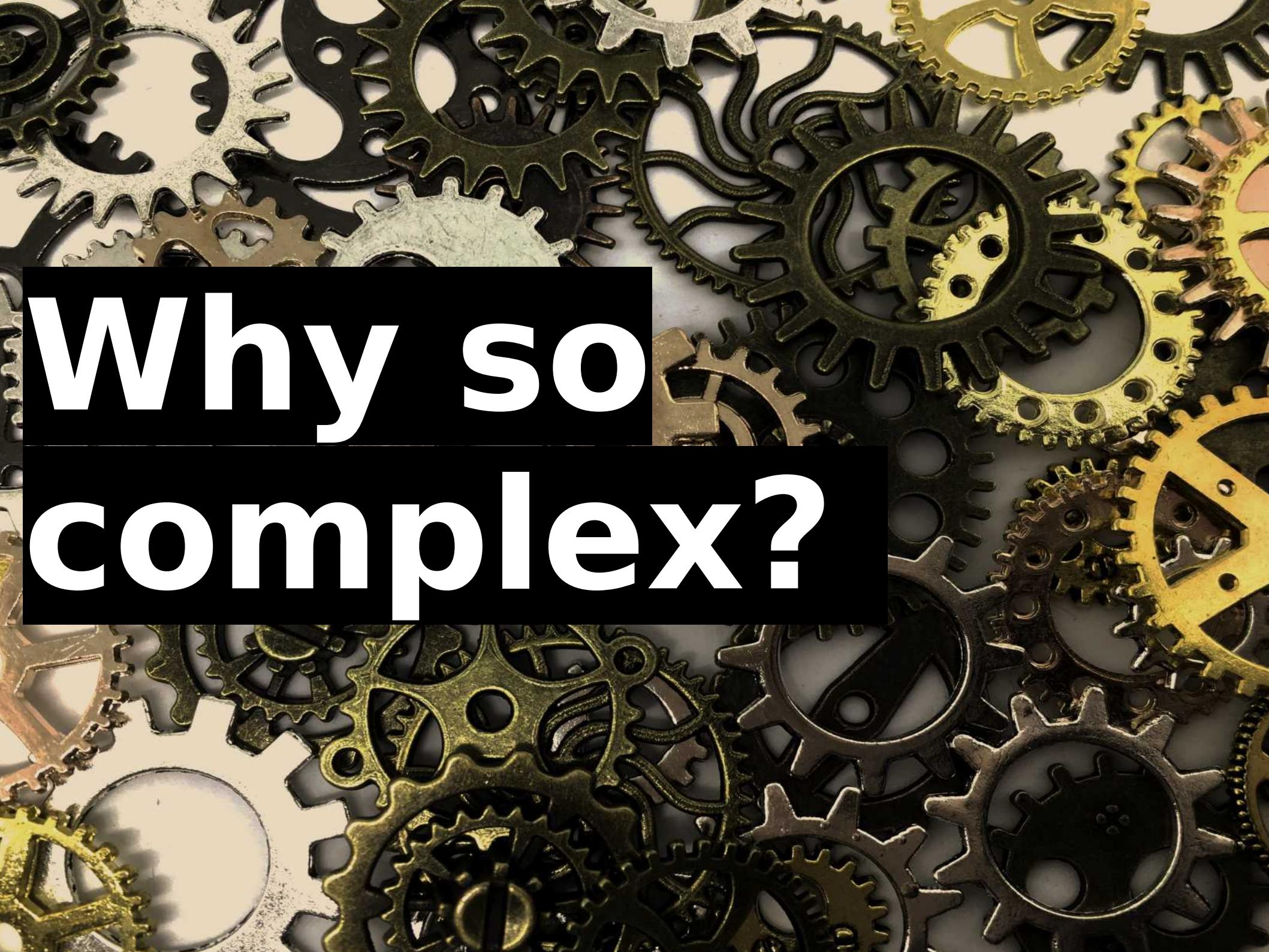
```
SOURCES = calculator.cpp add.cpp

OBJECTS = $(SOURCES:.c=.o)

calculator: $(OBJECTS)
           $(CXX) -o $@ $^
           -include $(SOURCES:.c=.d)

%.d: %.c
      @$(CC) -MM $(CPPFLAGS) &< | \
sed 's#\(\.*\)\.o: #1.o \1.d: #g' > $@
```

```
$ make  
cc -o calculator calculator.cpp \  
add.cpp mul.cpp sub.cpp div.cpp  
$ make  
make: 'calculator' is up to date.
```



Why so
complex?

Clean Build



```
SOURCES := calculator.cpp add.cpp  
sub.cpp mul.cpp div.cpp
```

```
PROGRAM := calculator  
LIBRARIES := libc libz
```

```
INCLUDE framework2.so
```

A photograph of various hardware tools on a white background. In the top right corner, a black-handled screwdriver with a silver tip is partially visible. In the bottom left corner, a single Phillips head screw lies horizontally. In the center, a larger screw is positioned vertically. A thick black rectangular box covers the central portion of the image, containing the text.

**Why are you
here?**

Survey



A photograph of a workshop or garage. In the foreground, a wooden workbench is cluttered with various tools and equipment, including a blue power drill, a black power tool, and a white cylindrical object. Behind the workbench, a brick wall is visible, showing signs of age and wear. A metal pipe runs along the wall. The lighting is somewhat dim, creating a focused atmosphere on the workbench area.

Build Errors

Bad

dependencies

Buggy software images



Slow compilation



THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."

HEY! GET BACK
TO WORK!

COMPIILING!

OH. CARRY ON.



Updating build description files



10%



65.00€



130.000€
/Year

A green apple is positioned behind a large kitchen knife, with its blade resting on the wooden surface. The knife has a black handle with three silver rivets. The background is a light-colored wooden texture.

130.000 p.A. * 10% *
10 Programmers =

130.000€



130.000 € / 220 days =

590 €

Loss/Day



SUPER SIZED
MEALS.com



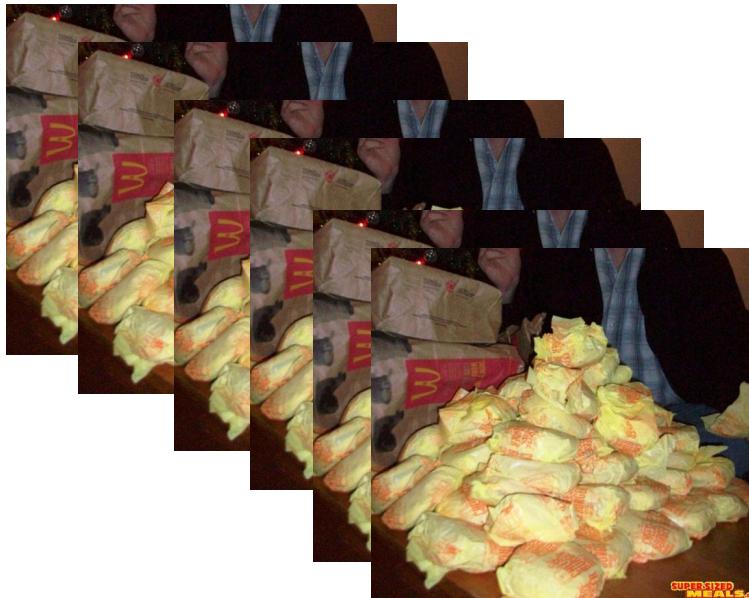


Holiday









A good Build System



A good Build
System
minimizes
Costs



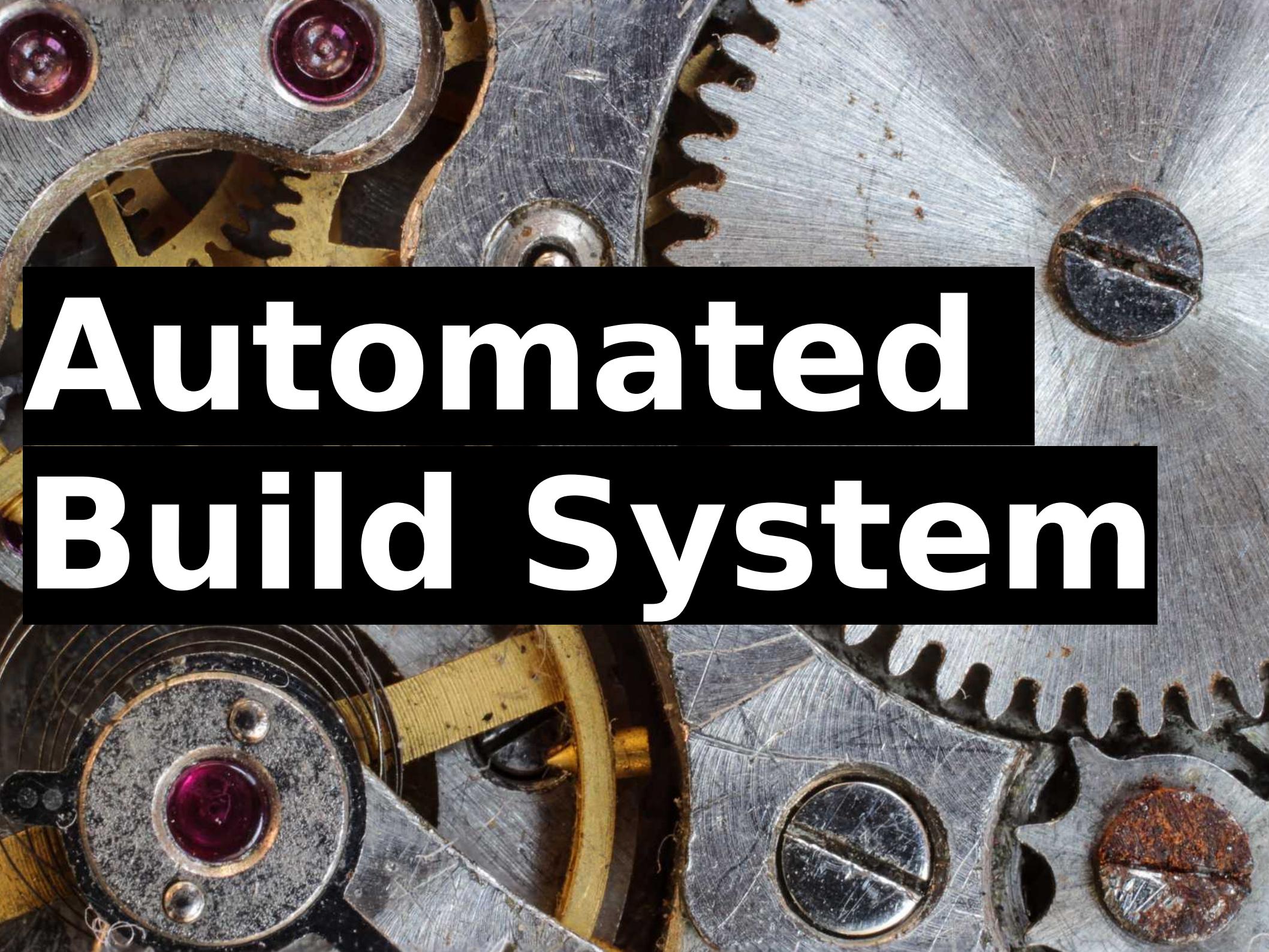
A composite image featuring a close-up of a man's face on the left and a wide shot of an industrial factory floor on the right. The factory floor is filled with complex machinery, conveyor belts, and robotic arms. A large black rectangular overlay contains the text "The Goal".

The Goal

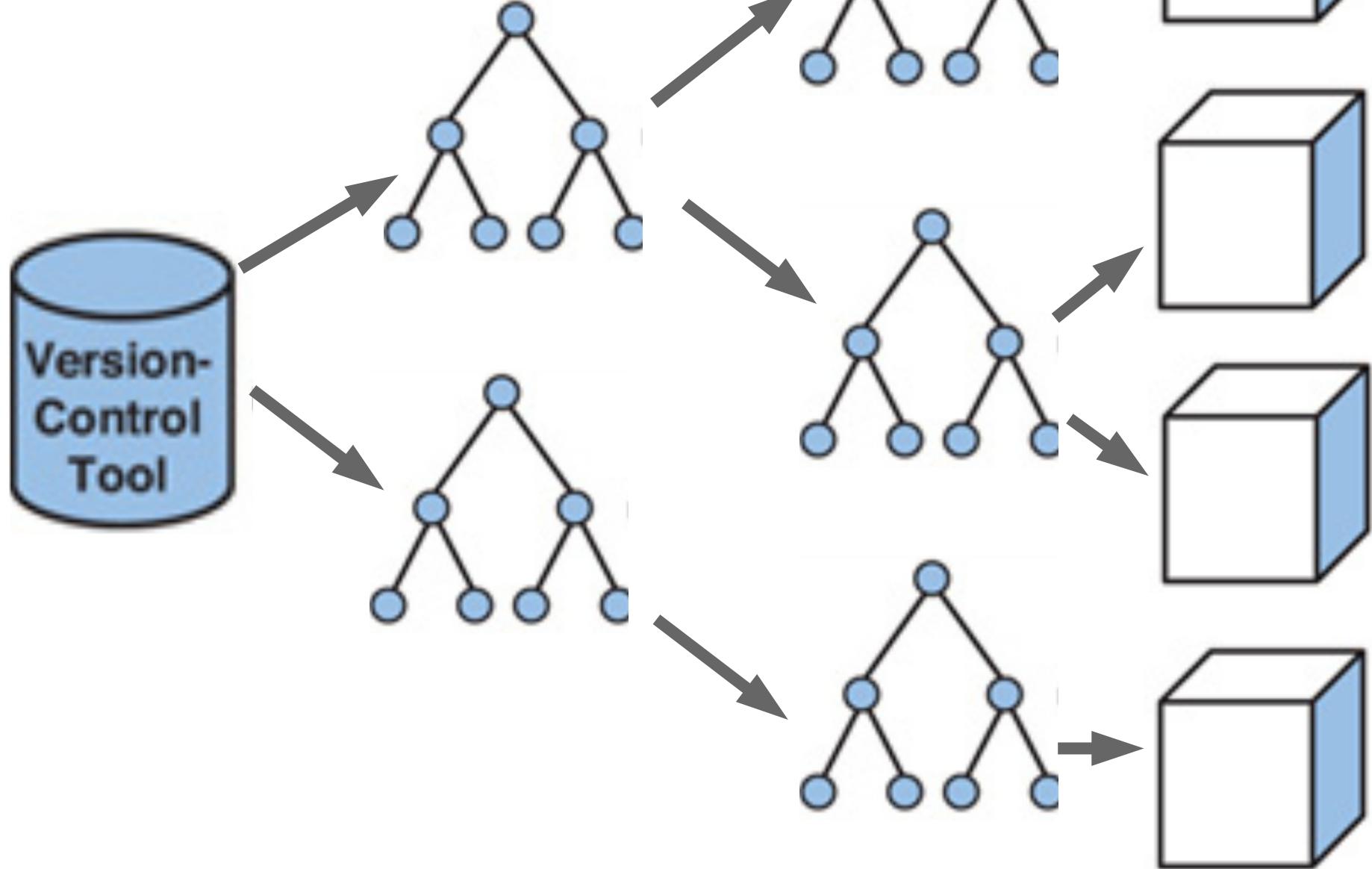
A composite image featuring a close-up of a man's face on the left side, looking towards the right. The background is a factory floor where a car chassis is being assembled on a conveyor belt. The text "Secondary Goal" is overlaid in large, white, sans-serif font on a black rectangular background that spans the width of the image.

Secondary Goal

Automated Build System



Multitarget



Compiled language

Compiled Language

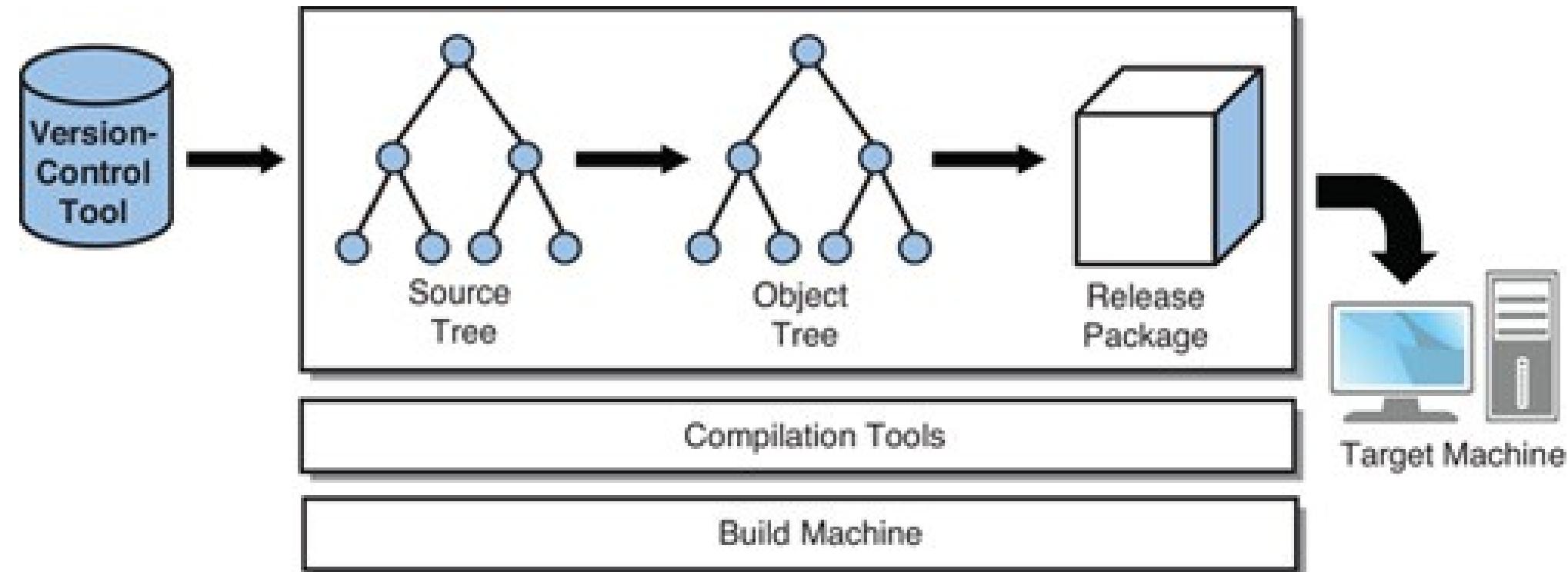


Image from 'Software Build Systems: Principles and Experience'

Static- Analysis Path

Static Analysis

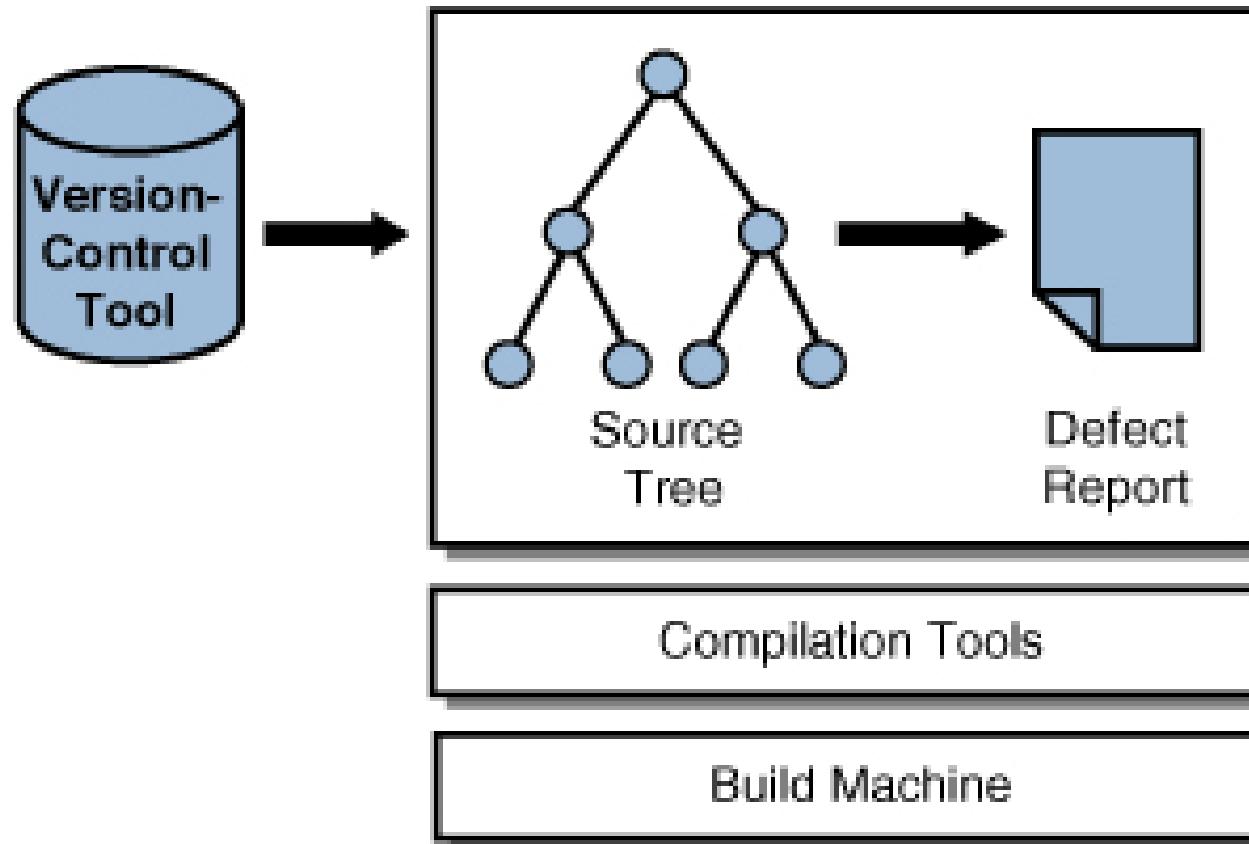


Image from 'Software Build Systems: Principles and Experience'

Document Generation

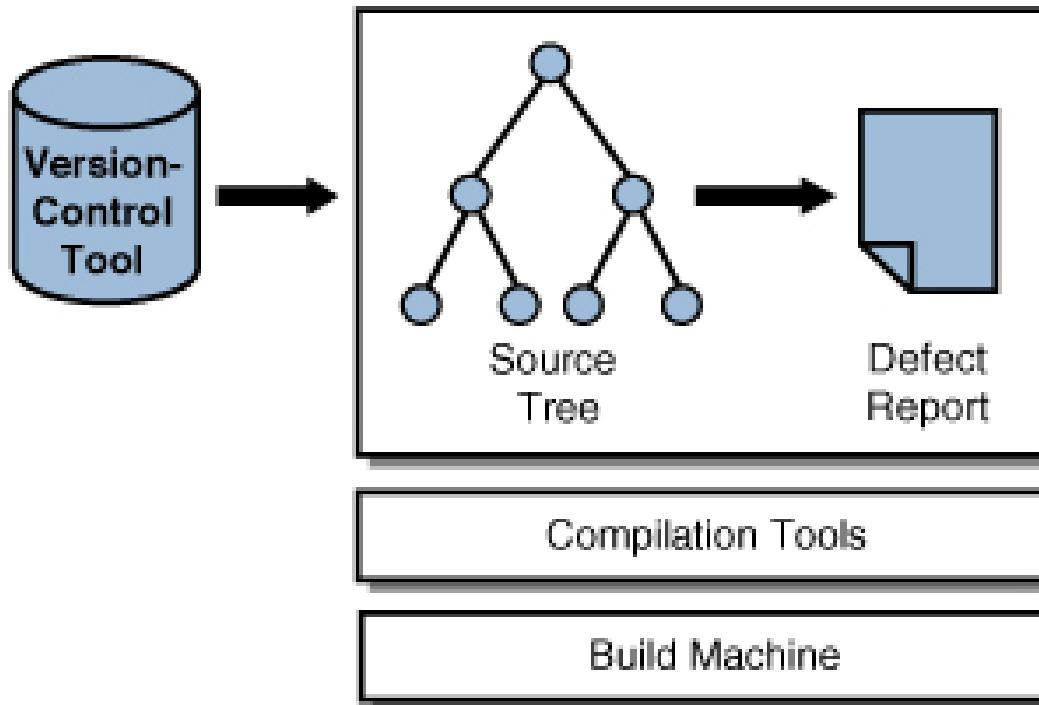


Image from 'Software Build Systems: Principles and Experience'

Build-Tool

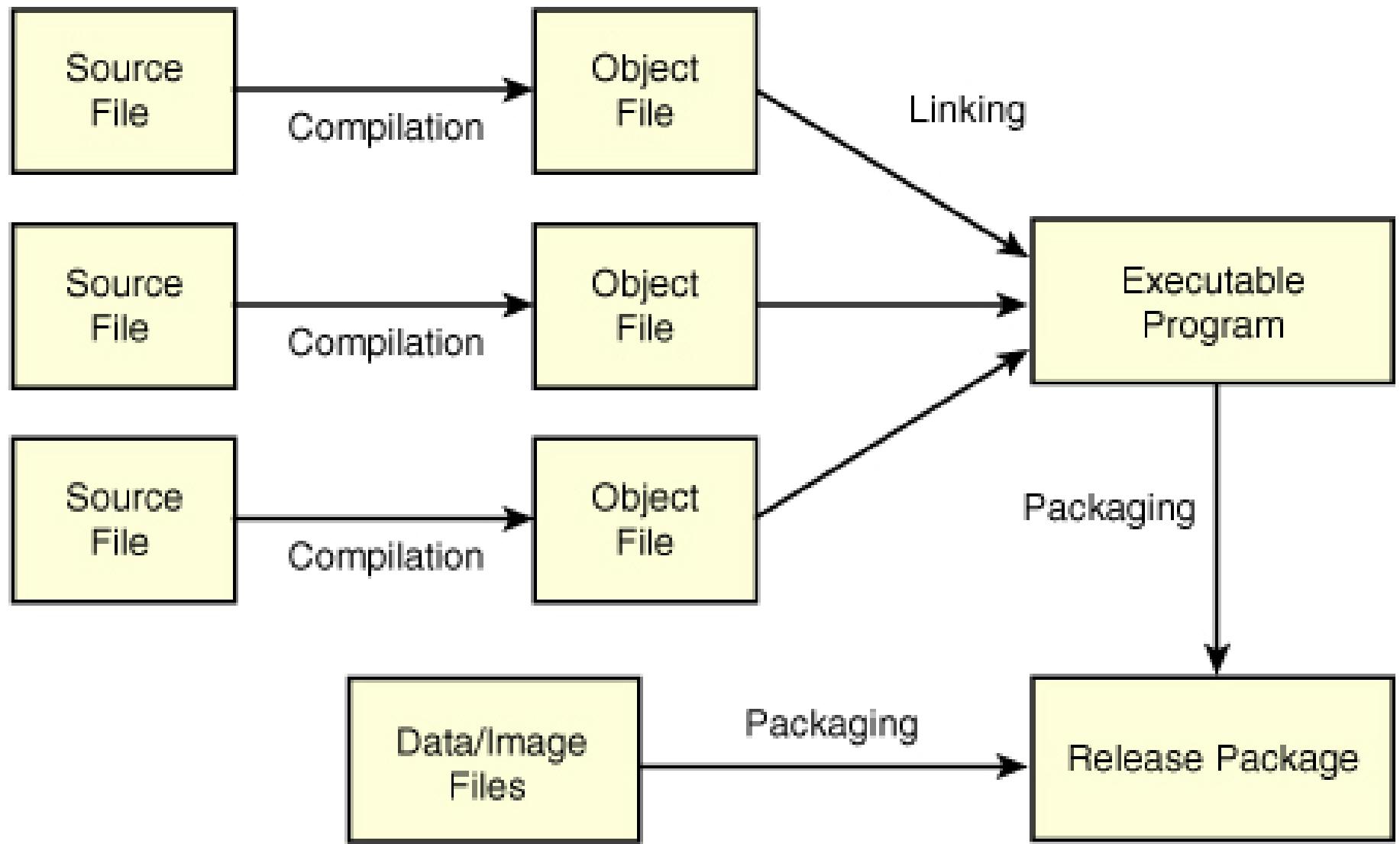


Image from 'Software Build Systems: Principles and Experience'

- * Developer build
- * Release build
- * Sanity build

- * Build-Management-Tool
- * Build-System
- * Build-Description
- * Build-Tool
- * Steps

CI-Pipeline

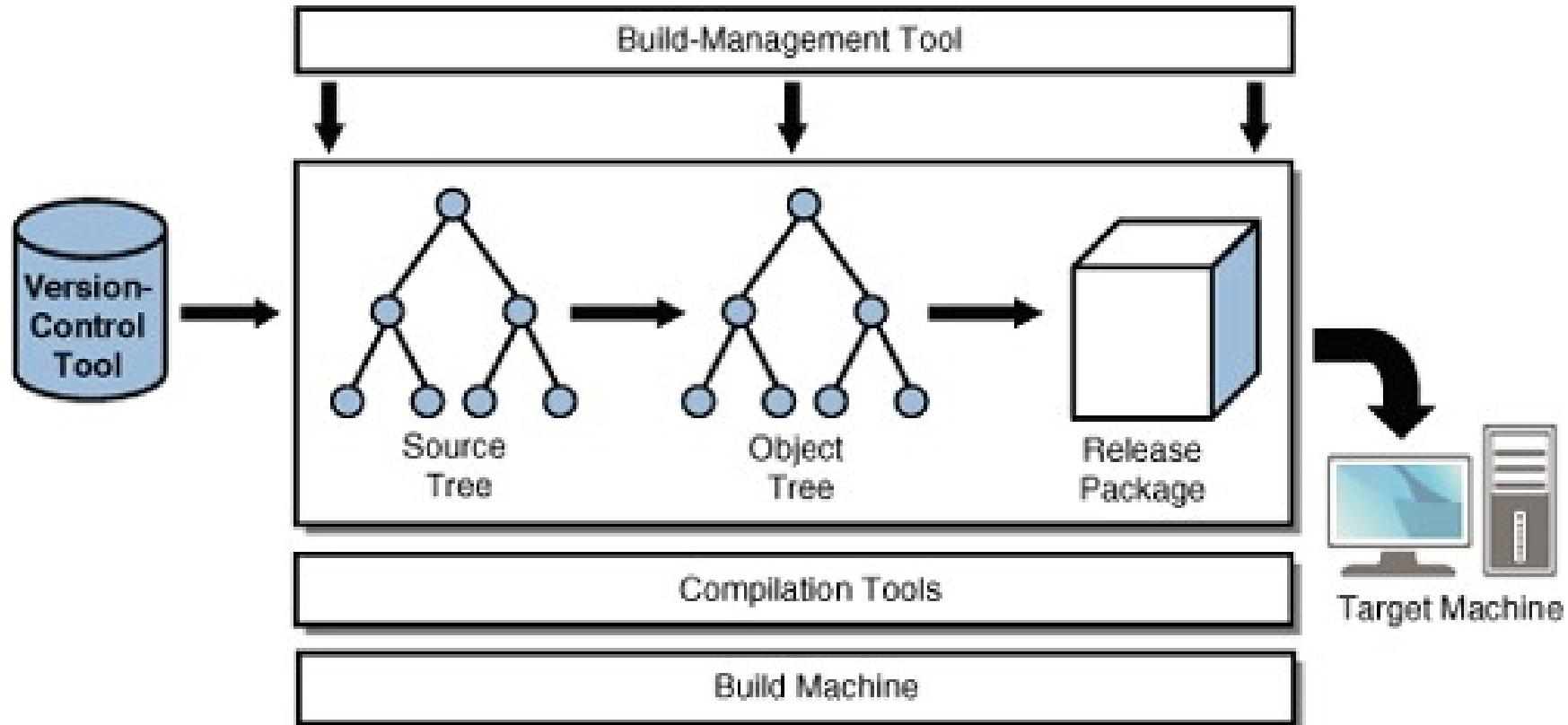
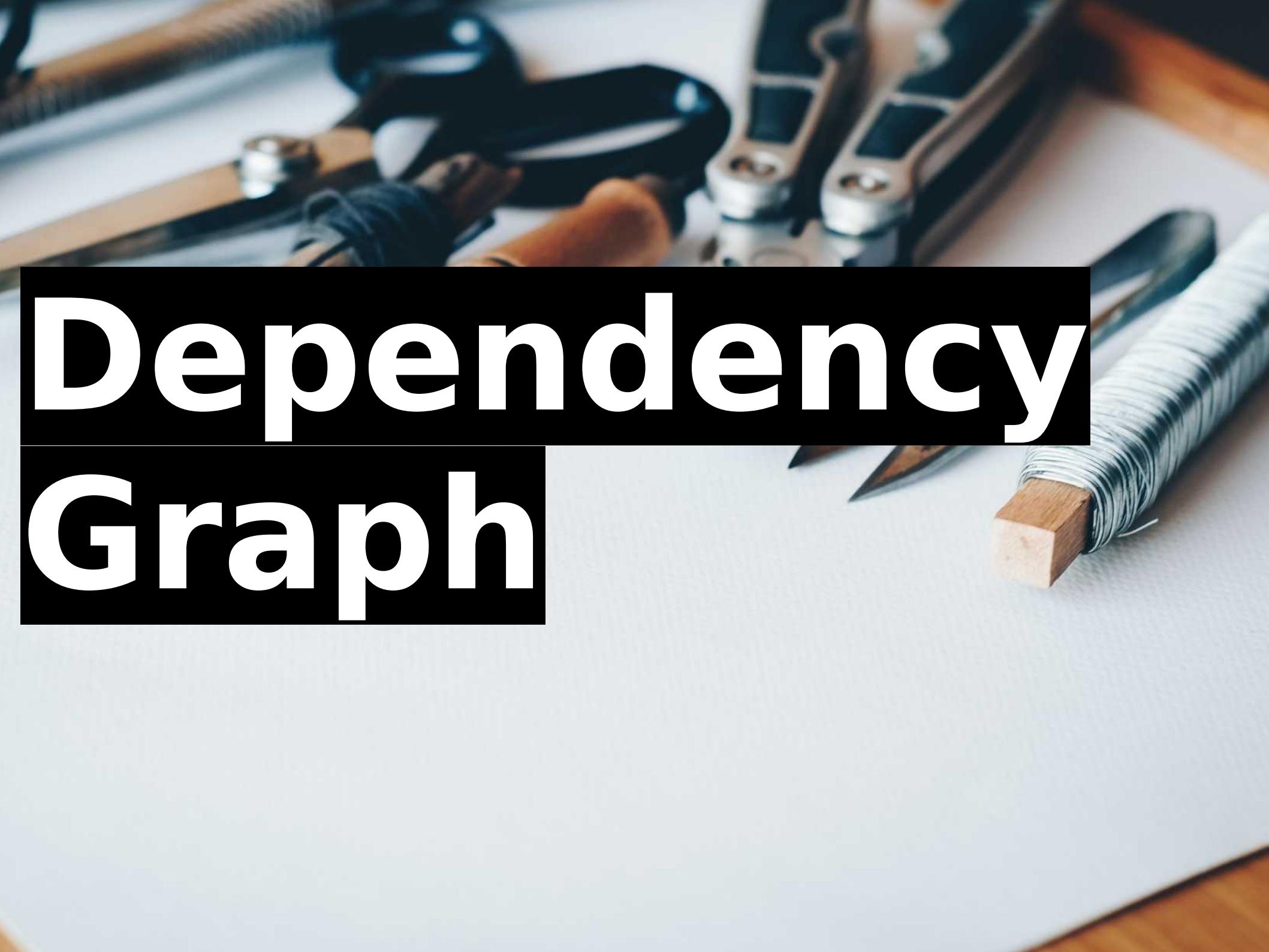
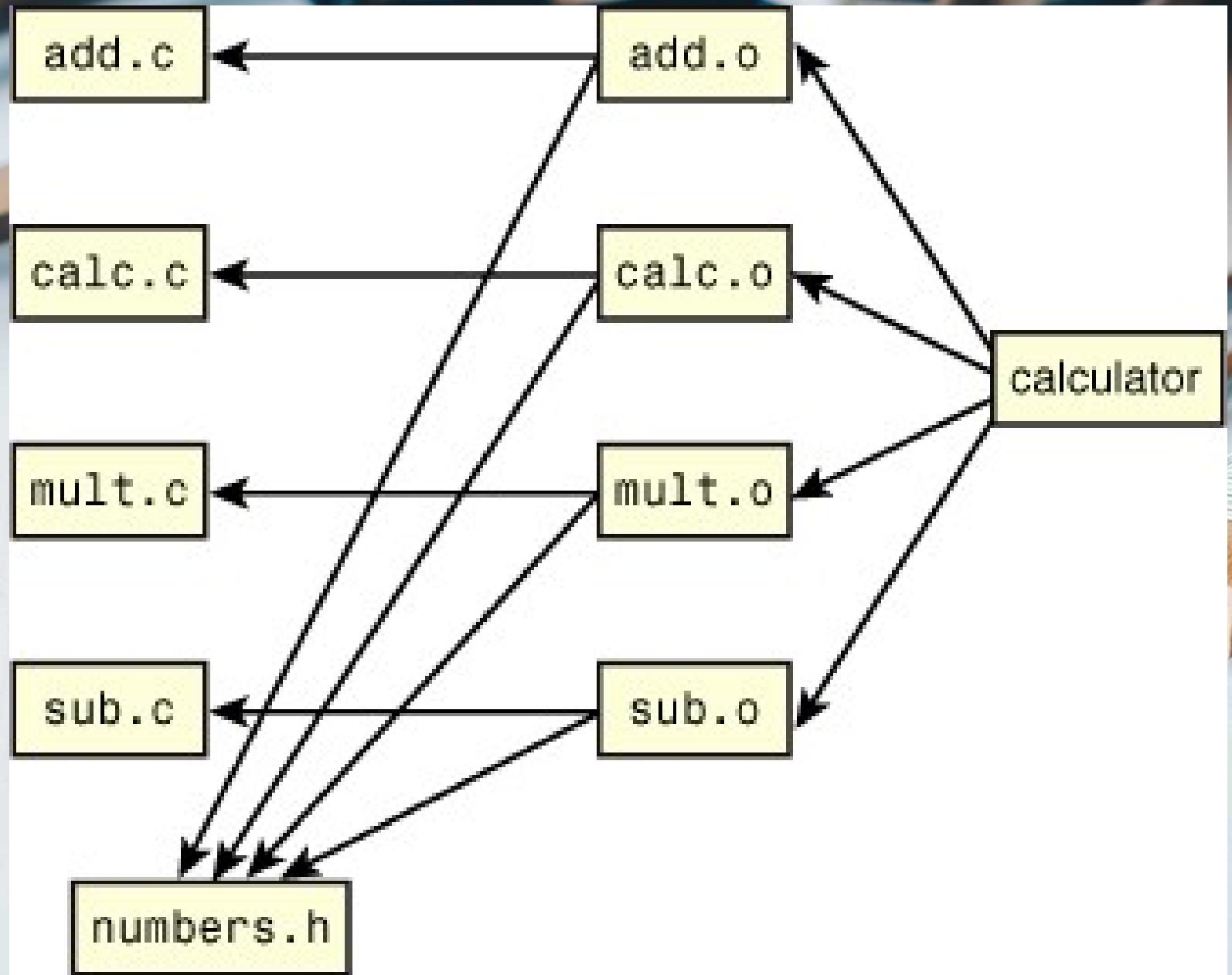


Image from 'Software Build Systems: Principles and Experience'

Dependency Graph





Metric



Subtargets & Build Variants



- 
1. Sub-targets
 2. Different editions
of the software
 3. Different target
architectures

Learnability



Declarative & Completeness



Convenience



Correctness



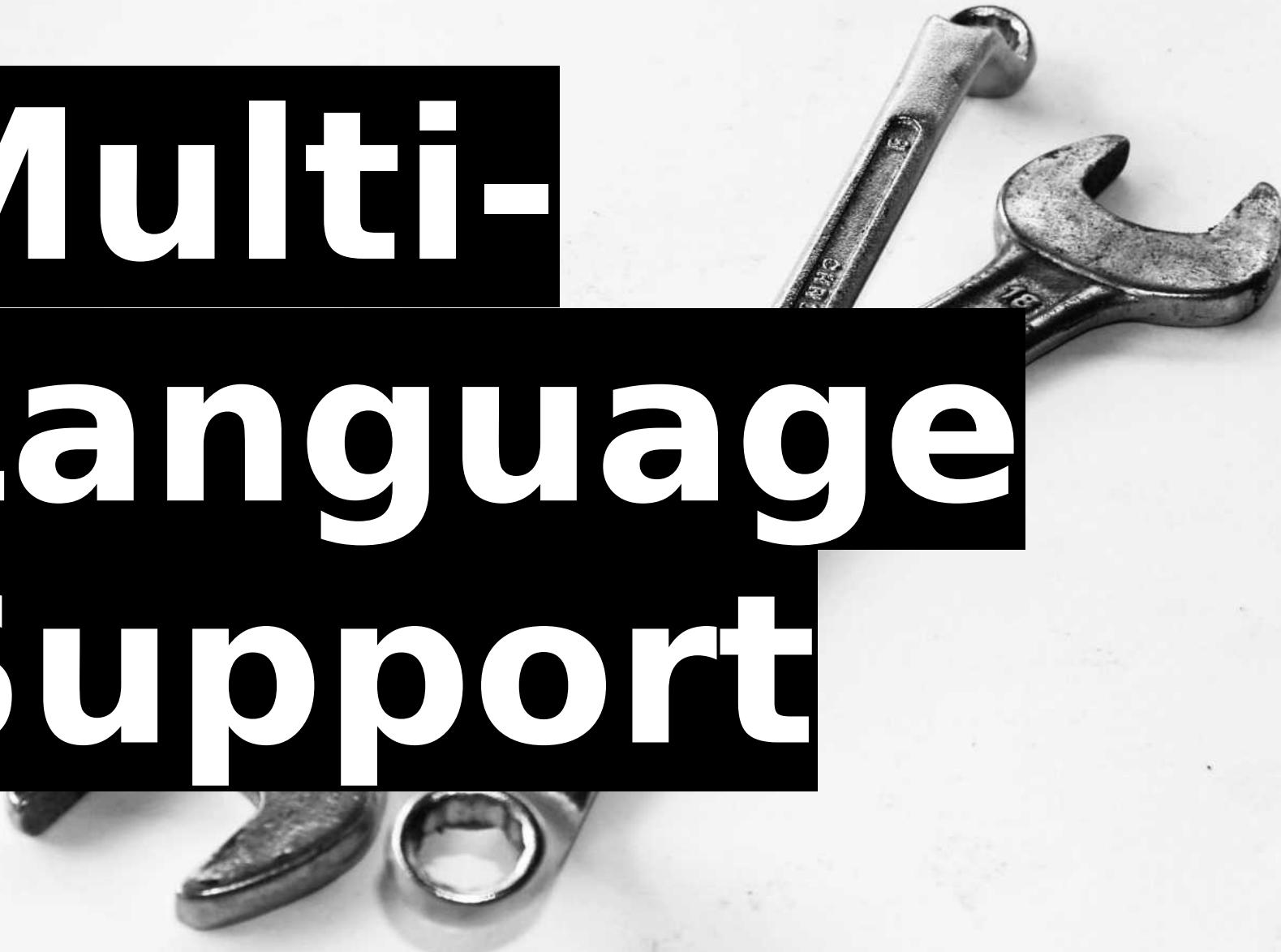
Performance



Scalability



Multi- Language Support



building sub-targets

Different editions of software

Different target architectures

Learnability

Declarative

Completeness

Convenience

Correctness

Performance

Scalability

Multi-Language-Support

S1: Source Code in a Single Directory

S2: Source Code in Multiple Directories

S3: Defining New Compilation Tool

S4: Building with Multiple Variants

S5: Cleaning a Build Tree

S6: Debugging Incorrect Builds



Build Tools

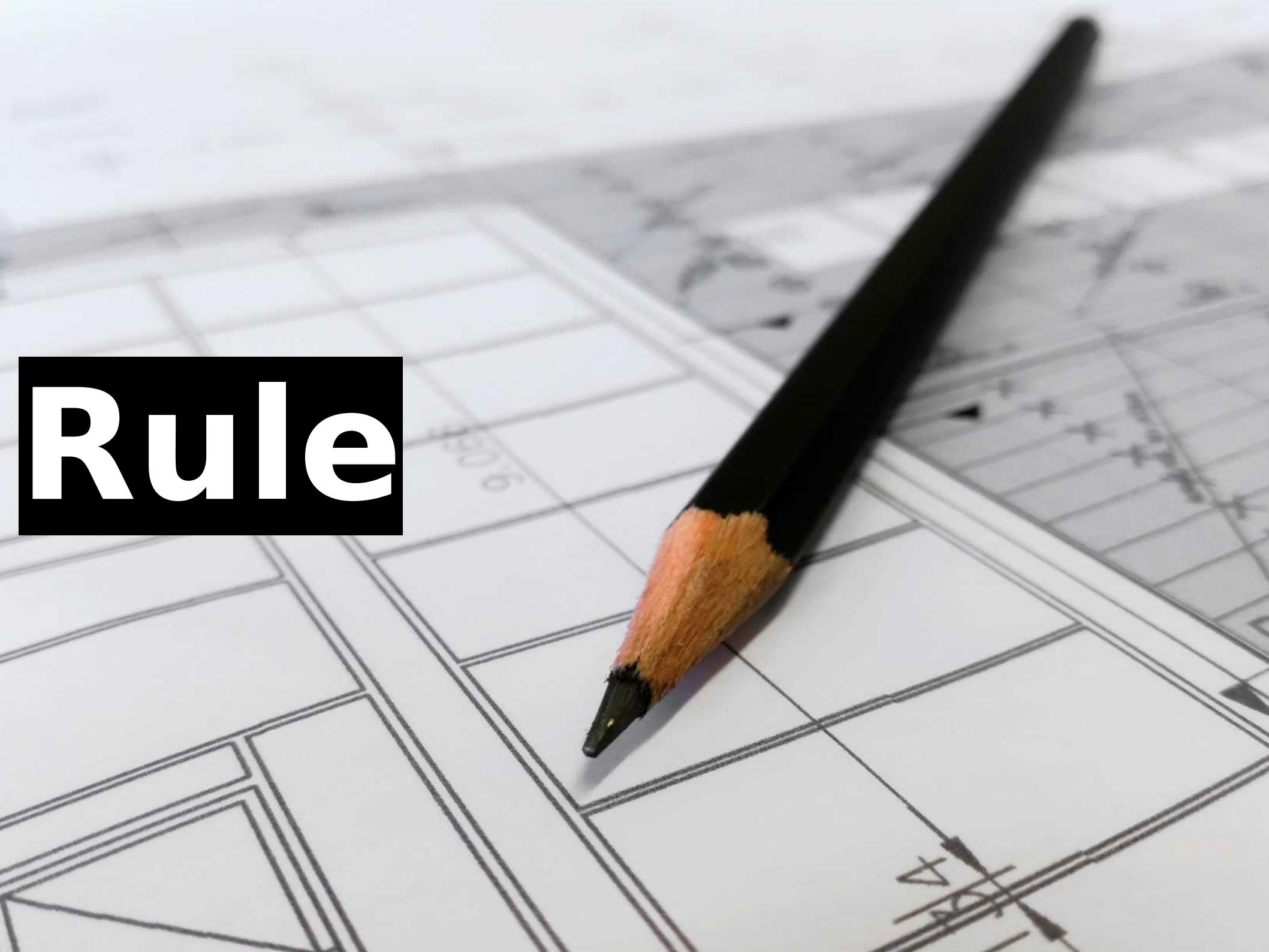




Make

<Low Level>

Rule



```
myprog: prog.cpp  
lib.cpp  
g++ -o myprog  
prog.cpp lib.cpp
```

building sub-targets	Yes
Different editions of software	Yes
Different target architectures	Yes
Learnability	Very Bad
Declarative	Yes
Completeness	Bad, But Turing complete
Convenience	Very Bad
Correctness	Bad
Performance	Good
Scalability	Good
Multi-Language-Support	Yes



<Alternative>
Ninja

<Low Level>



CMake

<Meta>

A close-up photograph of a large pile of rusty metal chain links. The chains are heavily coated in a reddish-brown rust, with some areas showing through where the metal has worn away. The lighting highlights the texture of the rust and the metallic surfaces of the links.

The Idea

A close-up, high-angle shot of a pile of rusty metal chain links. The chains are heavily coated in a dark reddish-brown rust, with some areas showing through where the metal has worn away. The lighting highlights the texture of the rust and the metallic surfaces of the links.

Turing Complete?

CMakeLists.txt



**Love it /
Hate it**

A close-up photograph of a large pile of rusty metal chain links. The chains are heavily coated in a dark reddish-brown rust, with some brighter orange-yellow areas where the surface has been partially removed or where light reflects off the edges of the links. The chains are intertwined in various directions, creating a complex, textured pattern.

Do it right

building sub-targets	Yes
Different editions of software	Yes
Different target architectures	Yes
Learnability	Bad , Because of Documentation
Declarative	Yes/No
Completeness	Good , Sometimes hard
Convenience	Good
Correctness	Very Good
Performance	Very Good
Scalability	Very Good
Multi-Language-Support	Bad , find_package problems



<Alternative>
Auto-
make

<Meta>



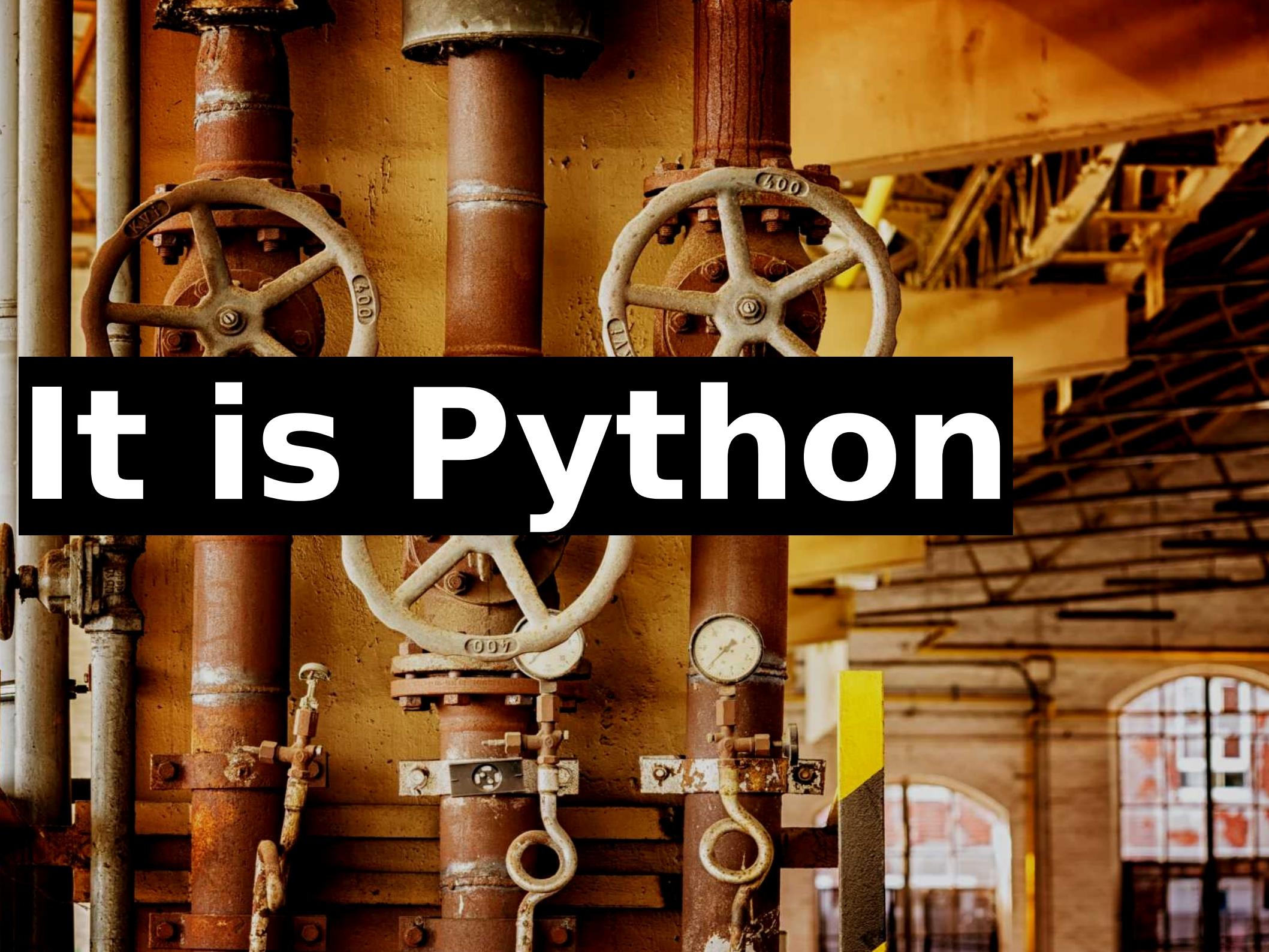
<Alternative>
Qmake

<Meta>

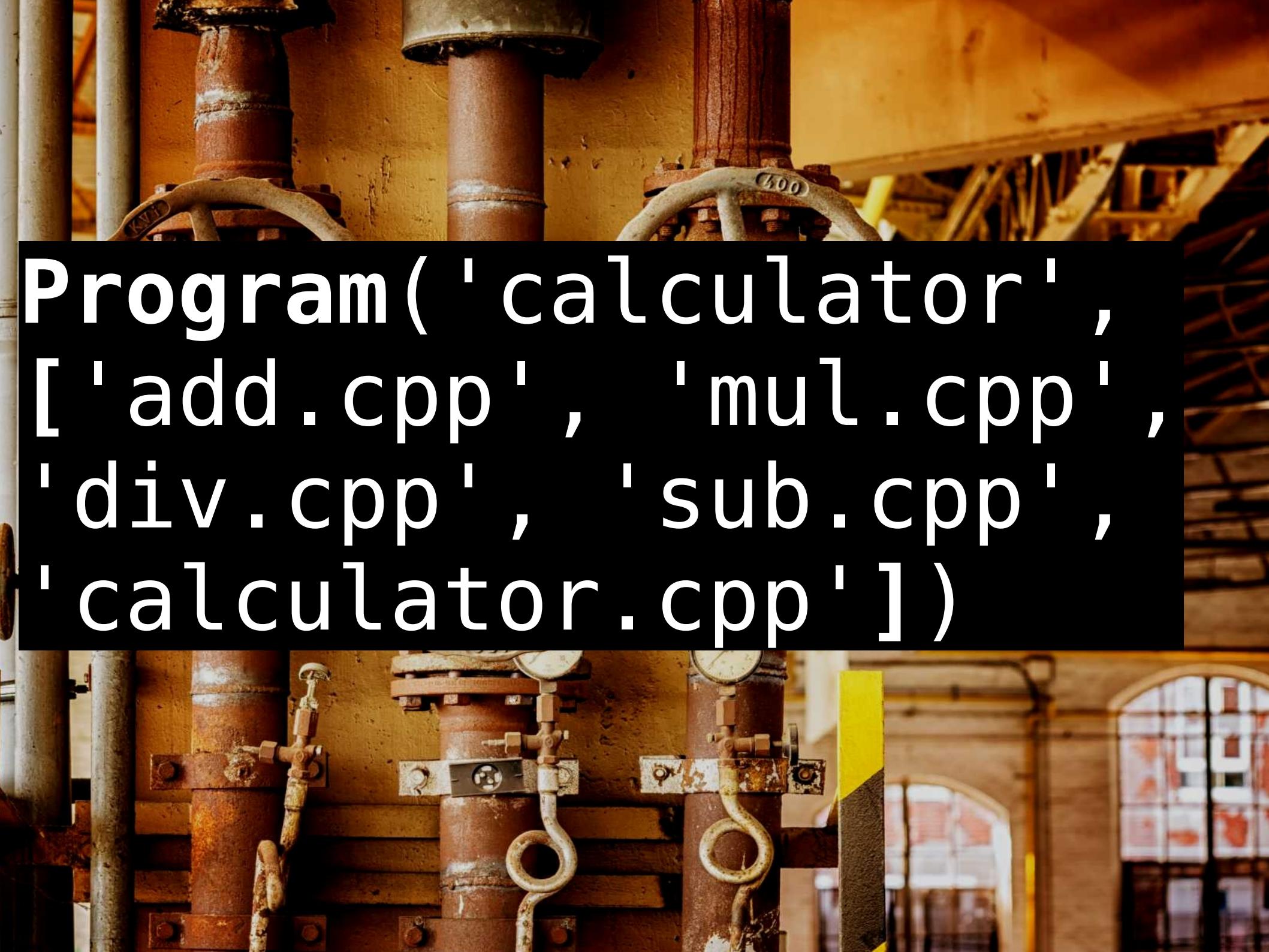


sCons

<Imperative>

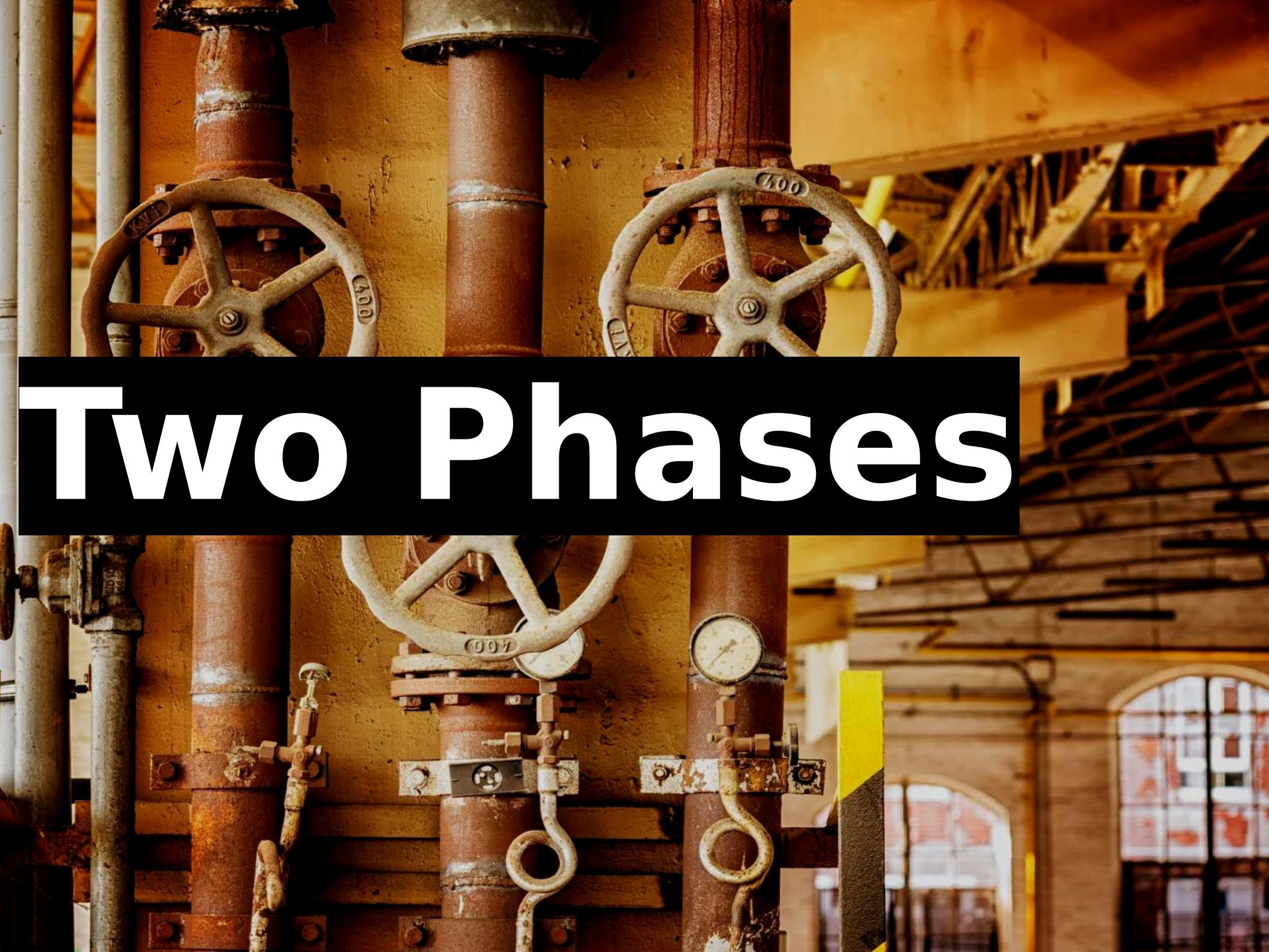
The background image shows a close-up view of industrial piping and valves. The pipes are made of a reddish-brown material, possibly cast iron or steel, and are connected by large, white-painted metal valves with circular handles. One valve has the number "400" printed on its handle. In the foreground, there is a complex arrangement of pipes, valves, and a pressure gauge. The scene is set in a factory or industrial facility, with wooden beams and ladders visible in the background.

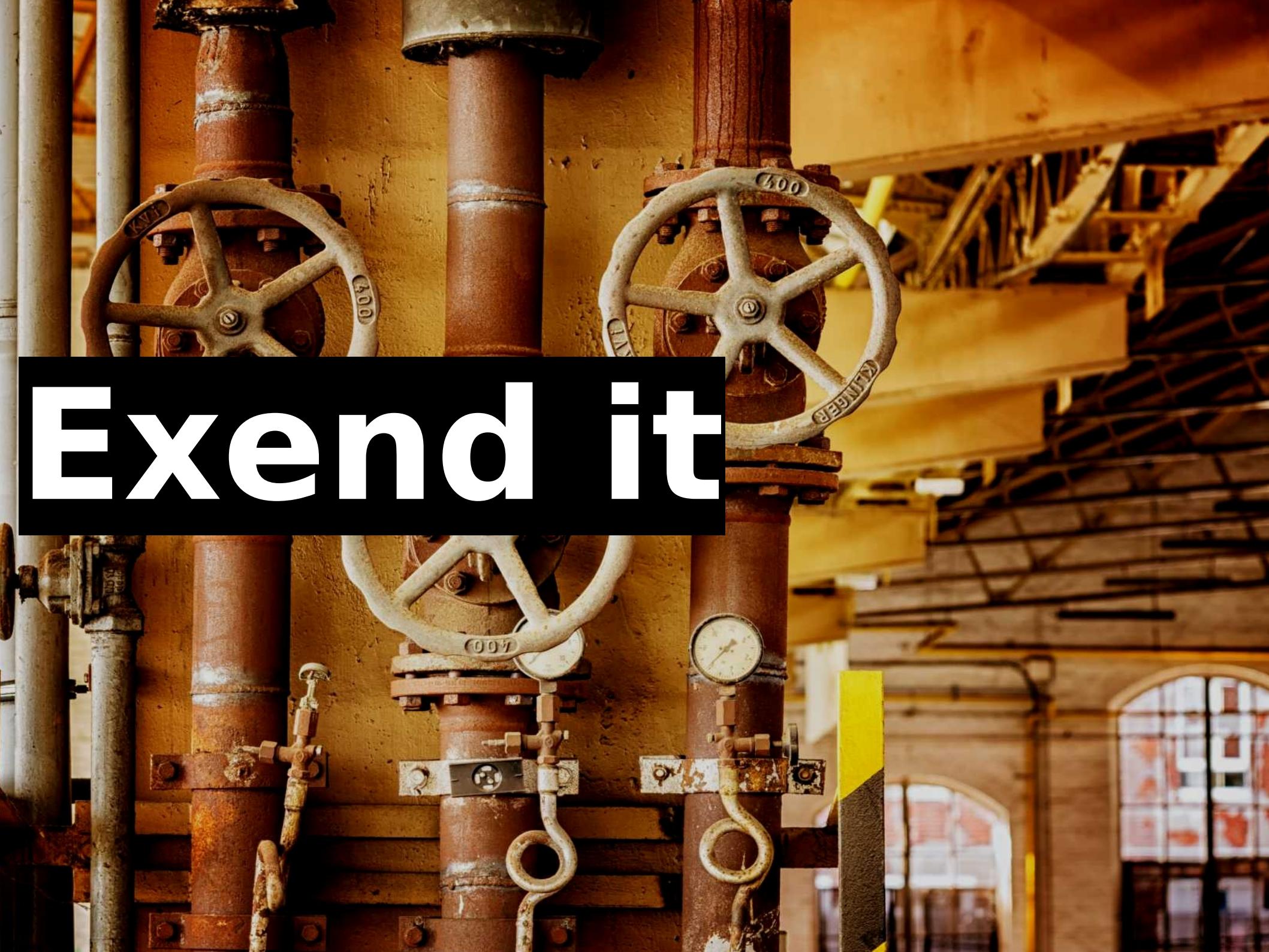
It is Python



```
Program('calculator',
['add.cpp', 'mul.cpp',
'div.cpp', 'sub.cpp',
'calculator.cpp'])
```

Two Phases

The background of the image is a collage of industrial photographs. It includes close-up shots of large, weathered metal valves with circular handles and markings like '007' and '400'. There are also vertical pipes, some with pressure gauges attached. The overall color palette is dominated by earthy tones like browns, yellows, and reds, giving it a historical and mechanical feel.



Exend it

building sub-targets	Yes
Different editions of software	Yes
Different target architectures	Yes
Learnability	Very Good
Declarative	No
Completeness	Very Good
Convenience	Very Good
Correctness	Very Good
Performance	Good
Scalability	? => Bad
Multi-Language-Support	Very Good



<Alternative>
Other

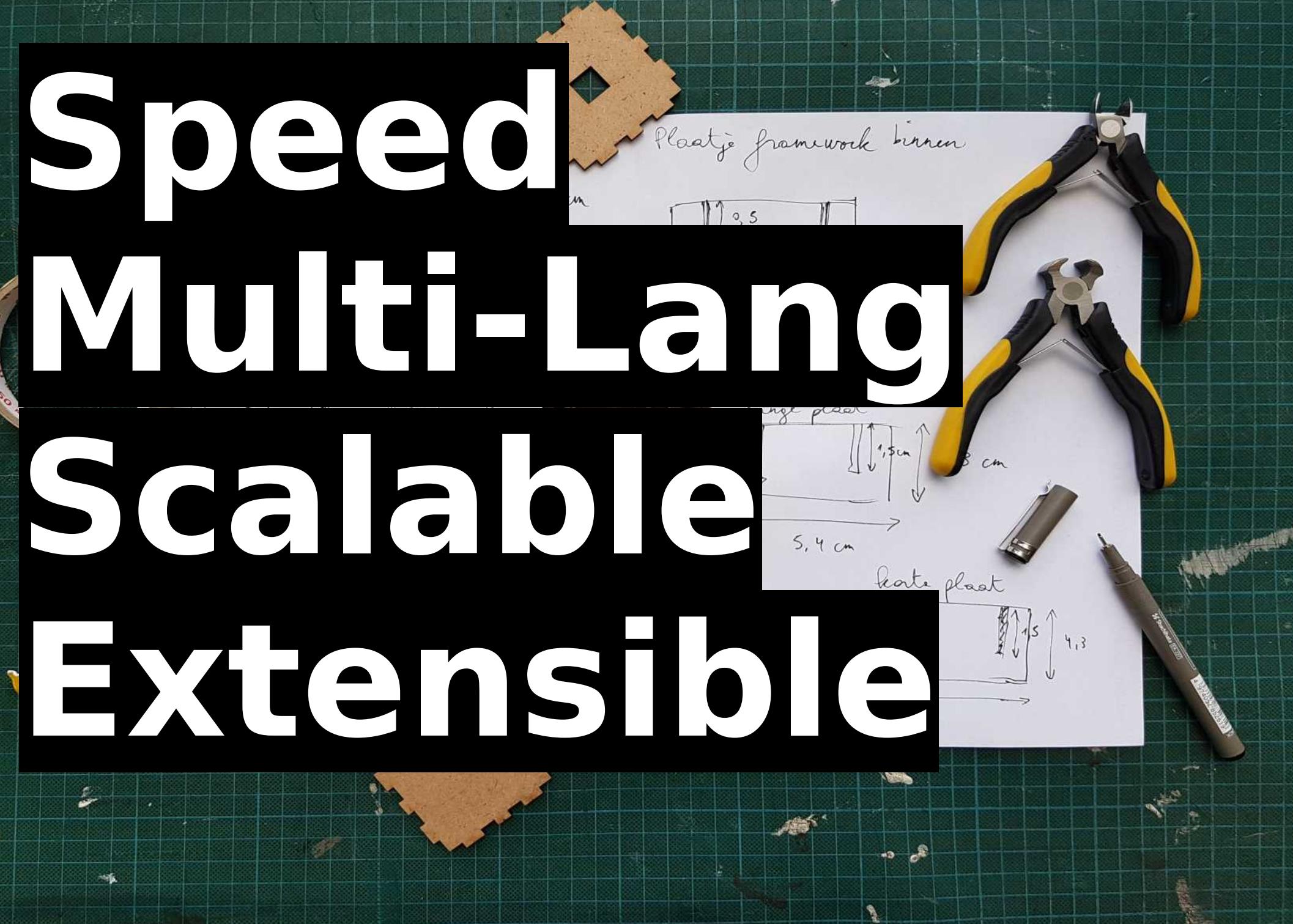
<Imperative>



Bazel

<Declarative>

Speed Multi-Lang Scalable Extensible



```
cc_binary(  
    name = "calculator",  
    srcs = [ "add.cpp",  
             "mul.cpp",  
             "calculator.cpp" ],  
)
```

Afmetingen button

BOVEN

Plaatje framework binnen

building sub-targets	Yes
Different editions of software	Yes
Different target architectures	Yes
Learnability	Very Good
Declarative	No
Completeness	Very Good
Convenience	Very Good
Correctness	Very Good
Performance	Good
Scalability	Very Good
Multi-Language-Support	Very Good



<Alternative>
Boost.
Build

<Declarative>



<Alternative> Gradle

<Declarative>



**<Alternative>
Buck**

<Declarative>



Custom

<DIY>

A photograph of a man from the chest down. He is wearing a light blue t-shirt and a tan apron with dark blue straps. Two pencils are pinned to the center of his apron. Several tools are visible in his pockets: a red utility knife in the left pocket, a pair of pliers or a wrench in the bottom left pocket, and a yellow utility knife in the right pocket. He is standing in what appears to be a workshop or garage, with shelves of tools and equipment in the background.

Don't do it

TODO:

**Dependency Management, Managing of
Metadata, Packaging and Installation,
Build-Machines, Platform
Independence, Fast Build-Times,
Correct Build, Caching,
Parallelization, Build-Variant
Management, Much Much more**

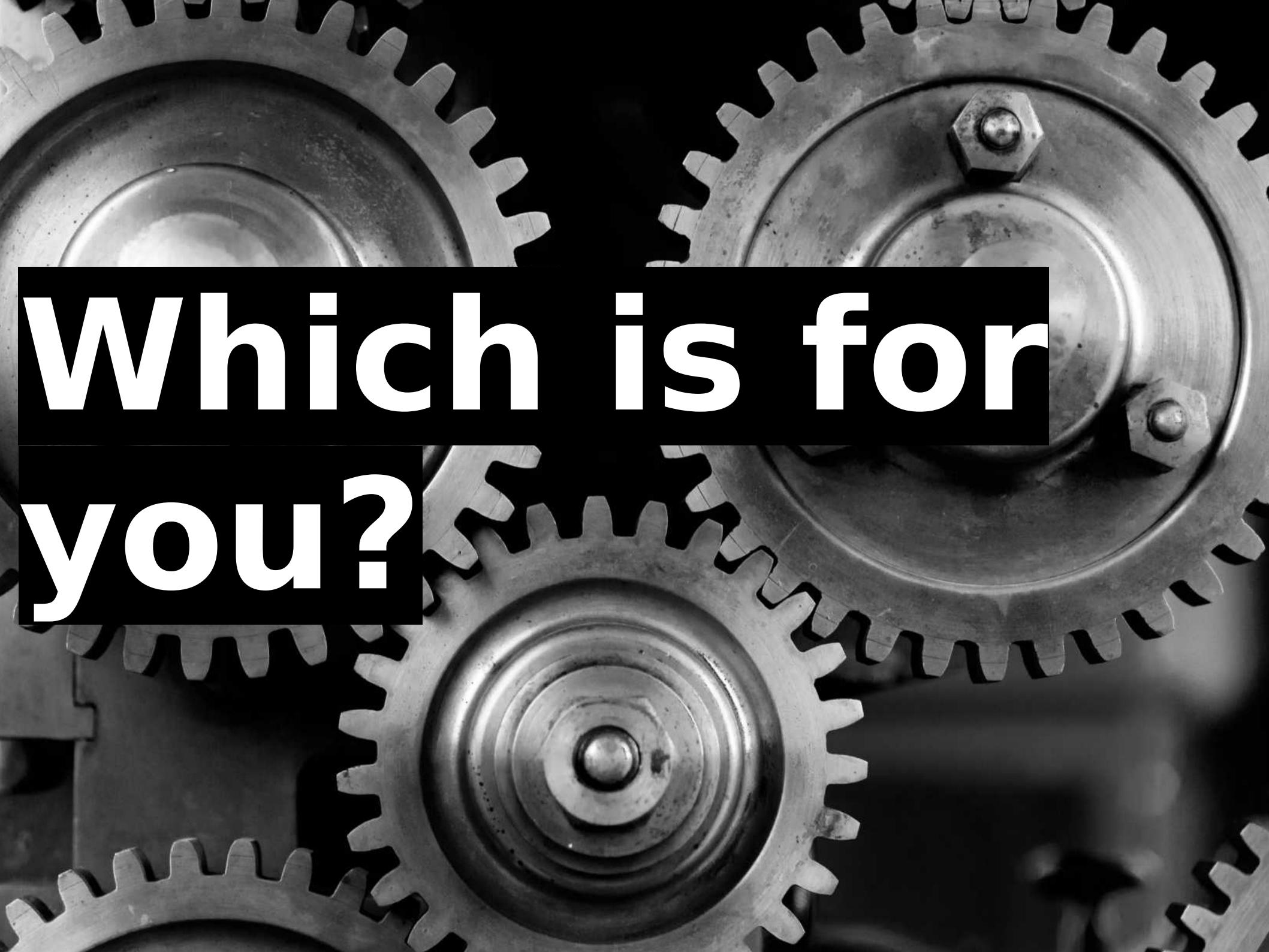


Software Build Systems

Principles and Experience



Peter Smith



**Which is for
you?**

- * Low Level
- * Meta
- * Imperative
- * Declarative
- * Custom



Thank
you!

Thomas Deniffel