MODUS CREATE

NAVIGATING COMPLEXITY

Scaling Software Development with Monorepos

Alois Cochard

October 27th 2023 | Microsoft Technology Center, Virginia

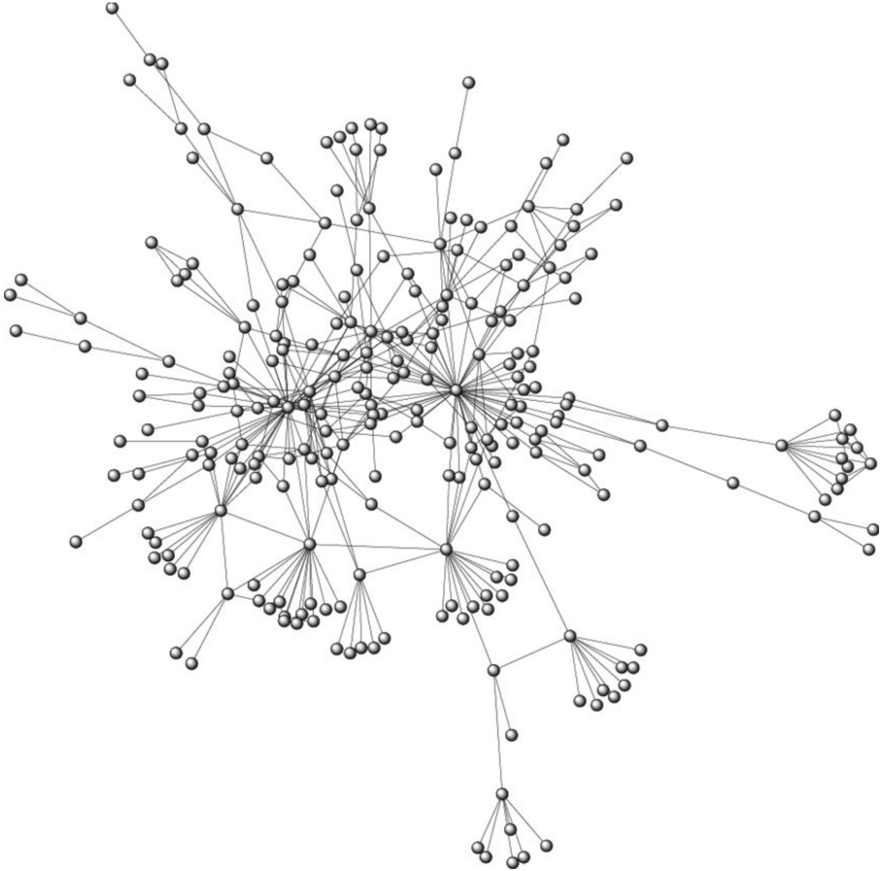
Software development is complex

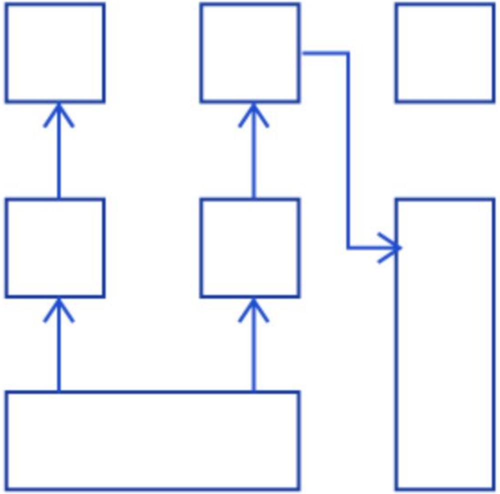
- Usage of third-party libraries
- Large code bases

Pain points when growing and scaling

- Dependencies – ensuring the same version is being used across all modules
- Collaboration issues – outdated branches and merging workflow cumbersome

We’ve all been here!



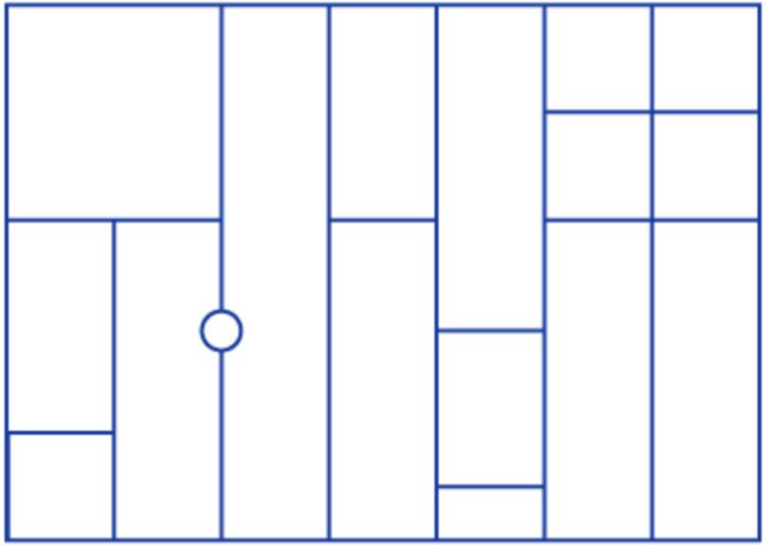


Do not fear complexity!

- Emerge naturally when solving hard problems
- Embracing Monorepos
  - Unified structure
  - Single point of entry

Hope to streamline software development process…

- Encourage collaboration
- Happy developers that are coding instead of fighting with processes
- Infrastructure as Code - encouraging all to be doing DevOps



Monorepos’ Developer Benefits

- Focus
- Taming complexity
- Accountability
- Reducing technical debt

Monorepos’ Scaling Benefits

- Collaboration
- Code sharing
- Unified dependency management

Companies using Monorepos

Google

Meta

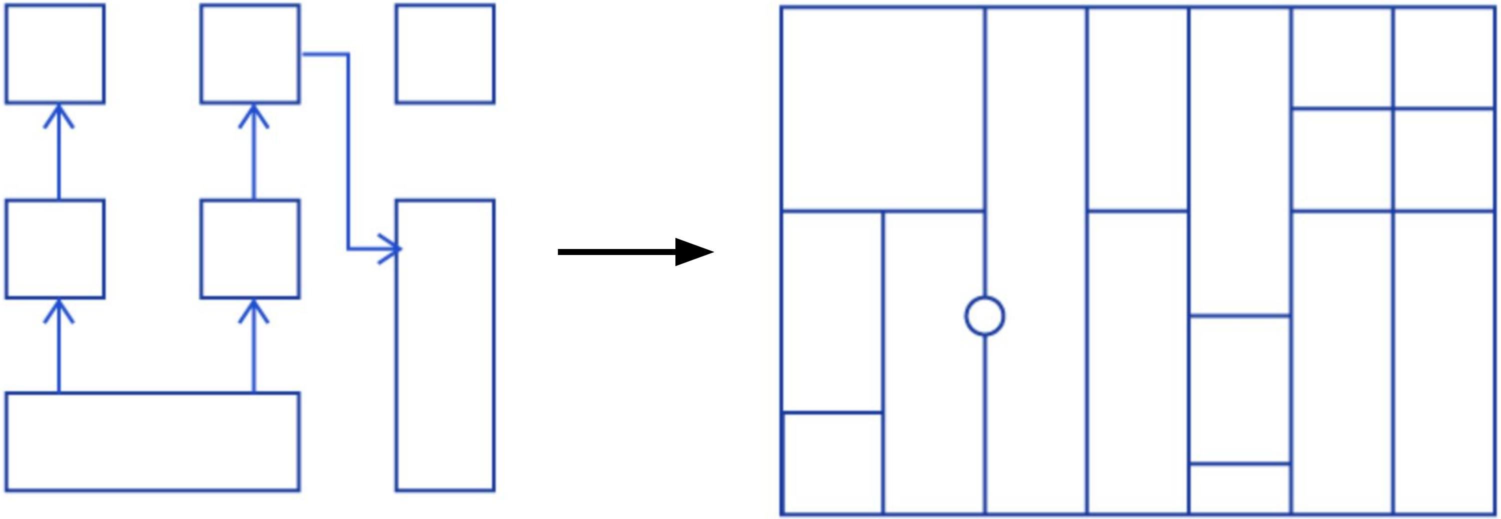
Microsoft

“It’s been a long road…” Maybe you heard those complaints?

- Checkout time - this is massive, especially for remote workers
- Build time - slow
- Tooling - hard to use
- Releases - releasing single component

Overcoming complexities and scaling challenges

- Checkout time
  - Tooling: Saping, Gerrit
  - Good practice: VCS Hygiene
- Releases
  - Infrastructure as Code
  - Good CI workflow
  - Good DevOps practices
- Build time
  - Good practices
  - Caching
  - Hemeticity
- Good tooling (innovation)
  - Bazel
  - Nix
  - Buck
  - Pants

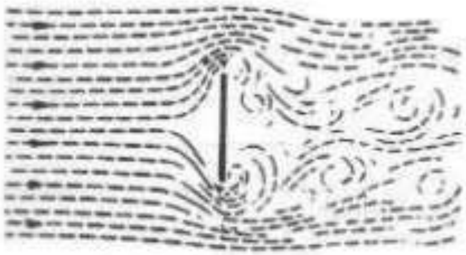


Implementing Monorepos at scale requires migration and must be well understood by the team.

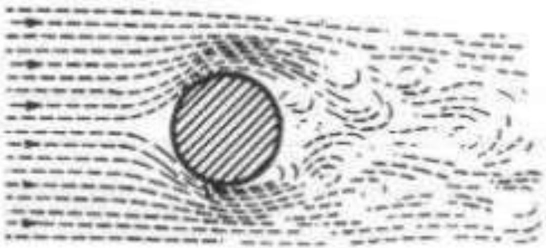
?

Who’s using or considering using Monorepos?

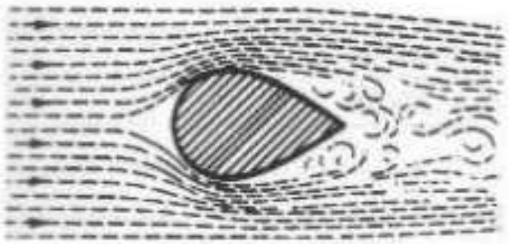
Resistance comes from heavy complex tool building. How do we reduce the resistance?



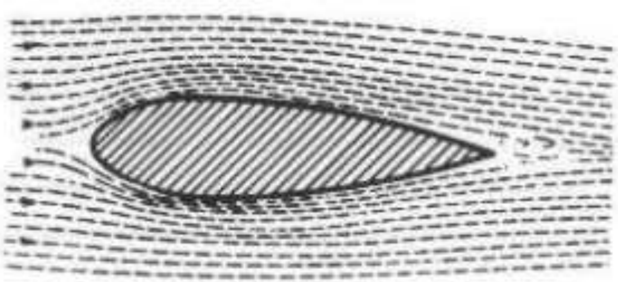
Resistance, 100%



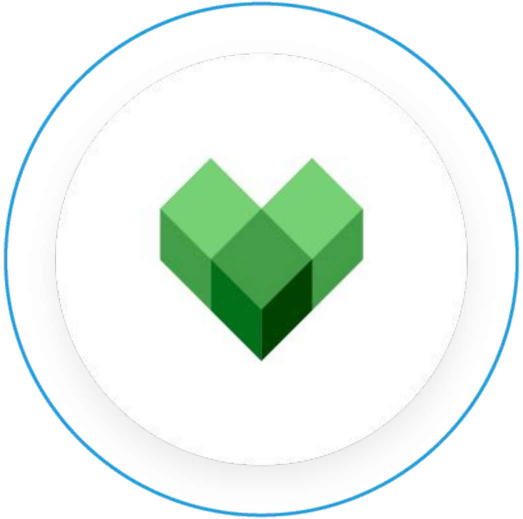
Resistance, 50%



Resistance, 15%

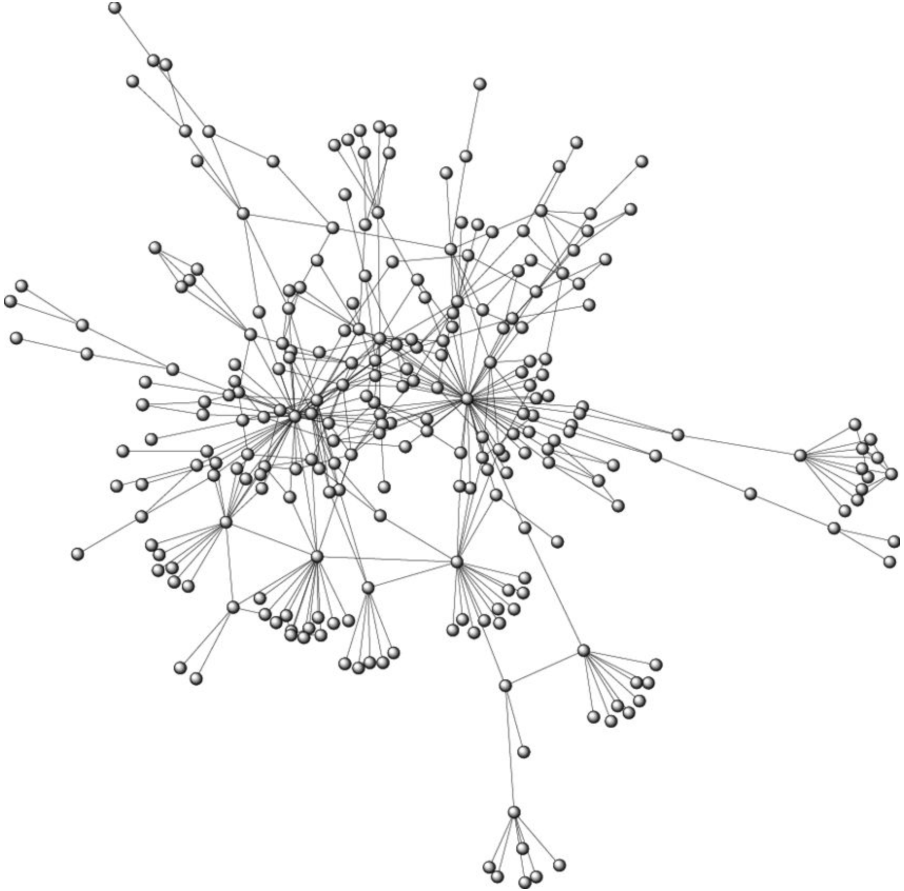


Resistance, 5%



Nix and Bazel - a unified way to manage dependencies and a unified development environment.

Java
Ruby
Python
Haskell
Go
PHP
language
Lua
JavaScript
Assembly
C
Groovy
Erlang
OCaml
Clojure
Scala
BASIC
Visual
AppleScript
Pascal
Scheme
Julia
Rust
Scratch
Language
Processing
LaTeX
Tcl
ColdFusion
APL
Ada
Delphi
F
IBM
REXX
DCL
ABC
Programming
TypeScript
Elixir
Eiffel
PowerShell
also
Lisp
E
PostScript
NET
Systems
Fortran
IMP
DASL
from
RPG
Common
C++
Basic
ISO/IEC
Robot
EXEC
ALGOL
KRL
JScript
Prolog
Logo
Script
Object
Simula
Dart
PL/SQL
now
DataFlex
Microsoft
Forth
FoxPro
Max
Bash
CPL
MATLAB
Smalltalk
shell
programming
Squeak
Not
CIL
J++
ksh
PL/I
CL
Q
of
G
L
A



Key Takeaways

- Monorepos are the most reasonable approach we know today to tackle large and complex projects
- There are many successful stories of companies scaling using monorepos
- It’s a lot about the culture now that we have good tooling

MODUS CREATE

THANK YOU!

Let’s make something great together.