

Angular Schematics

A Better
Developer
Workflow

Matt Vaughn
@angularlicious
matt@angularlicio.us

hi!



I am Matt Vaughn

Developer, Speaker, Consultant, PodCaster, Musician, Owned by Lukka



@angularlicious



github.com/angularlicious



<http://www.angularlicio.us> **OR** www.angularlicious.com

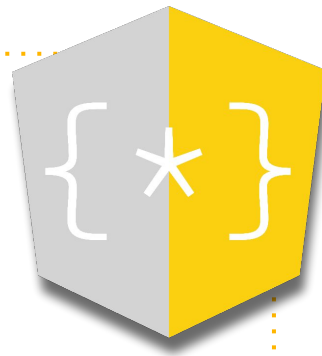
Lukka **THE** Husky

Profile:

- 4.5 years old
- 83 pounds
- Instagram: **lukka_the_husky**

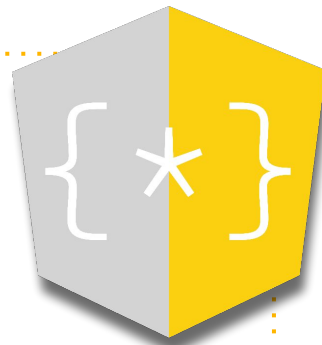


Presentation Resources



- **Angularlicious :: Learn Schematics**
 - [Github Workspace](#)
 - [Adding Schematics to Angular Workspace](#)
 - [How to Debug Schematics?](#)
- **Online**
 - [Building Schematics in a Sandbox](#)
 - [Introduction to Schematics by Hans](#)
 - [Unit Testing Schematics](#)
- **Guides**
 - [How to Add Schematic Projects to the Angular Workspace \(or Nx\)](#)
 - [Getting Started with Schematic Options](#)
 - [Add/Debug Schematic Projects in an Angular Workspace](#)
- **Book**
 - [Schematics: Generating Custom Angular Code with the CLI](#)
- **Videos**
 - [ngAir #134: Angular Schematics with Mike Brocchi](#)
 - [Custom Schematics by Manfred Stayer](#)
 - [ngHouston: CLI Schematics with Hans Larsen](#)

What really **MATTERS**?



At the end of day, what really matters...is **value**.

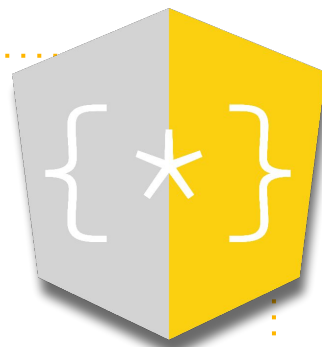
- Did we solve the problem?
- Is it a better experience?
- Is it more efficient?
- Did we reduce cost?
- Did it improve performance?
- Did it generate revenue or profit?
- Did it reduce risk?
- Did it improve someone's life or make them happy?
- Are you a happy developer?



Why workflow matters?

WHAT IS THE STATE OF DEVELOPER WORKFLOWS?

WHAT is the state of my **WORKFLOW?**

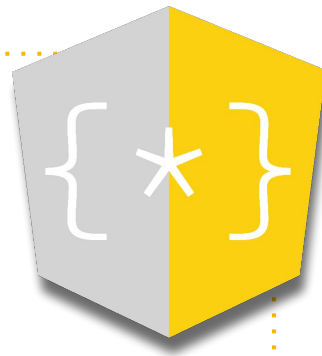


Many developers, teams, and companies are accelerating their workflow and delivery of high-quality solutions by taking advantage of mature:

- conventions and opinionated frameworks
- monorepos
- code generation and scaffolding
- packages, control suites, and tools
- accelerated processes
- language capabilities

Why pay **attention**?

Software development and technology in general is very **competitive**. If we are not **effective** as developers, teams, and companies - we will not last.



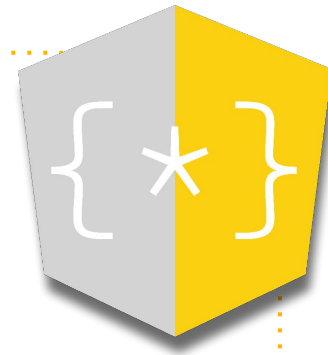


CODE

generators

Code Generation...

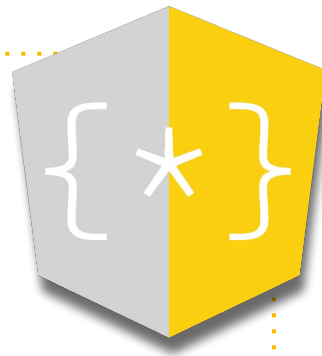
Why generate **code**?



Use tools to automate the creation of code that is difficult, error prone, complex, or repetitive - with a goal or objective of providing consistency and easier maintenance.

- Automate creation of routine or mundane coding tasks.
 - Boilerplate code
- Follows a specific recipe/template
 - Consistent
 - Maintainable
- Simplifies complex processes
- Less error prone - reduces human error
- Enforce standards, convention, compliance, and styles

How do we get to a point where **code generation is possible?**



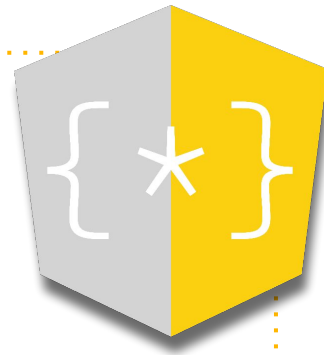
- Adoption of good programming practices and principles.
- Consistent, reliable, and conventional architecture using established patterns and practices
- Better frameworks
- Better tools with plug-in capabilities.
- Strongly-typed languages with Object-oriented capabilities.

What do
SCHEMATICS
give us?

?

DO I REALLY NEED SCHEMATICS?

Capabilities of Schematics



Project

- Scaffold new projects
- Configurable with options

Features

- Modules
- Components
- Services
- Classes, Enums
- Pipes
- Directives
- Routing
- Data Store
- State Management

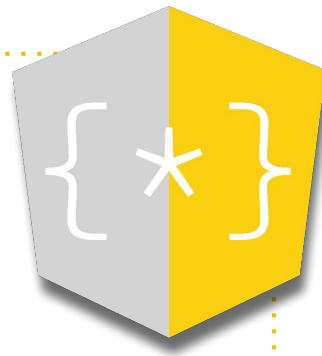
Tooling

- Code Formatting
- Testing
- Routing
- Package Updates
- Code Updates
- Configuration
- Execute internal and external schematics

Framework

- @angular/material
- @angular
- @ngrx
- @nrwl
- @akita
- etc..

How do **SCHEMATICS** work?



Input + (options) → Schematic **Engine** → **PROCESS**

- Applies *rules, templates, filters, etc.*

Output → folders and files

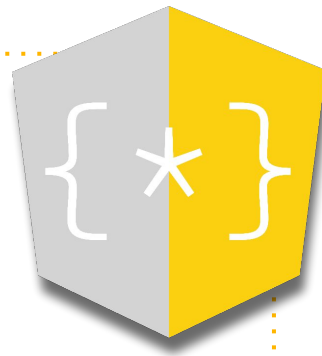
- Adds files, updates files, removes files



Getting STARTED

GETTING STARTED WITH SCHEMATICS IS EASY...HOW EASY IS IT?

Getting Started



Getting started demos:

- Environment and tools
- Building and testing
- Using options
- Templates
- Debugging
- Live execution
- Publishing



```
1 npm install -D @schematics/angular
2 npm install global @angular-devkit/schematics
3 npm install global @angular-devkit/schematics-cli
4 schematics --help
5 schematics --list-schematics
```

```
1 npm install -g @nrwl/schematics @nrwl/nx
2 create-nx-workspace learn-schematics --npm-scope=angularlicious
```

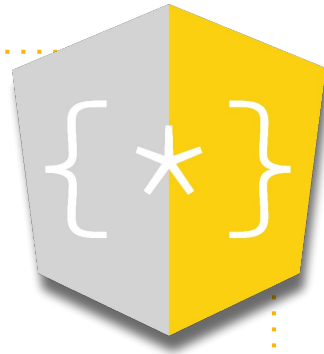


Final Thoughts

The future's so bright...

Final THOUGHTS

- Increases developer workflow capabilities
- More conventional and consistent
- Uses best practices and patterns
- Extensible



“

*BE SO GOOD
THEY CANNOT
IGNORE
YOU.*

Presentation **RESOURCES**



- **Angularlicious :: Learn Schematics**
 - [Github Workspace](#)
 - [Adding Schematics to Angular Workspace](#)
 - [How to Debug Schematics?](#)
- **Online**
 - [Building Schematics in a Sandbox](#)
 - [Introduction to Schematics by Hans](#)
 - [Unit Testing Schematics](#)
- **Guides**
 - [How to Add Schematic Projects to the Angular Workspace \(or Nx\)](#)
 - [Getting Started with Schematic Options](#)
 - [Add/Debug Schematic Projects in an Angular Workspace](#)
- **Book**
 - [Schematics: Generating Custom Angular Code with the CLI](#)
- **Videos**
 - [ngAir #134: Angular Schematics with Mike Brocchi](#)
 - [Custom Schematics by Manfred Stayer](#)
 - [ngHouston: CLI Schematics with Hans Larsen](#)

Thanks!



Any questions?

- Twitter: @angularlicious
- Email: matt@angularlicio.us
- Web: www.angularlicio.us
 - Web
 - Podcast
 - ebook