



SCHEMATICS for a Better Workflow

hi!



I am Matt Vaughn

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



@angularlicious



github.com/buildmotion

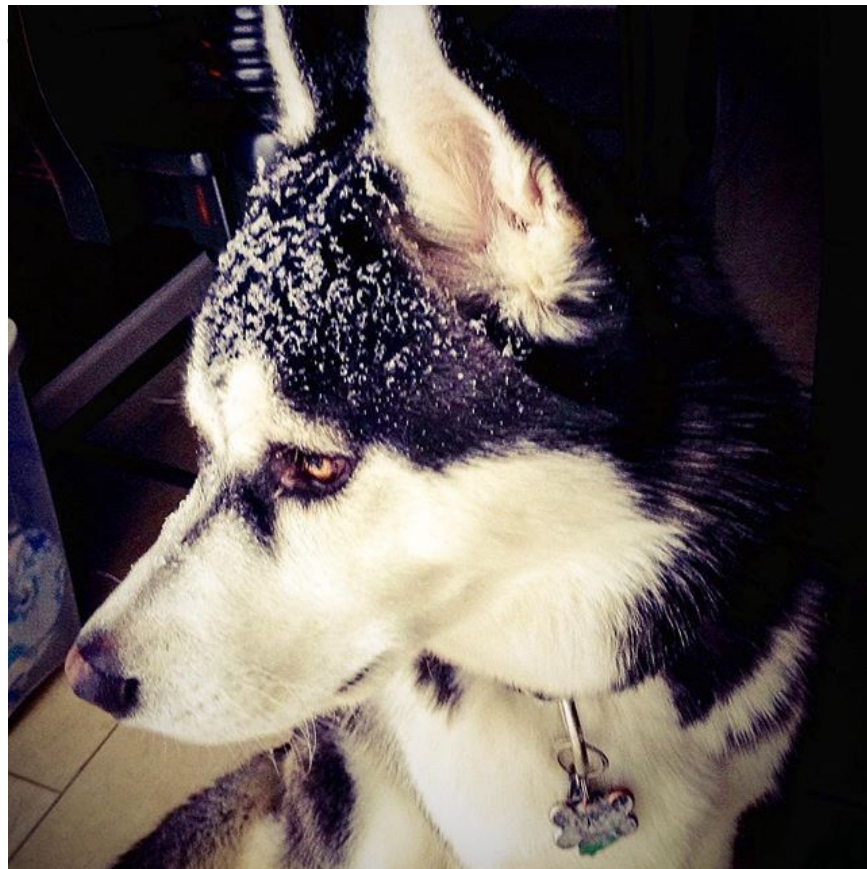


<http://www.angularlicio.us> **OR** www.angularlicious.com

Lukka **THE** Husky

Profile:

- 4 years old
- 80 pounds
- Instagram: **lukka_the_husky**



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 does
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 *action, templates, filters, rules, 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:

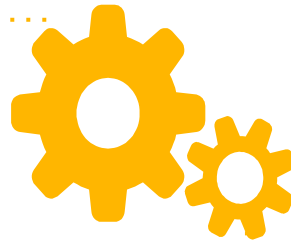
- Environment and tools
- Building and testing
- Using options
- Templates
- Debugging
- Live execution
- Publishing



Final THOUGHTS

The future is 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](#)
- **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