

# BUILDING ENTERPRISE APPLICATIONS WITH ANGULARJS

Christian Janz (@c\_janz)

[christian.janz@bridging-it.de](mailto:christian.janz@bridging-it.de)





# Christian Janz

*Consultant im Bereich Softwareentwicklung  
Java/JEE, Web bei **Bridging IT** in Mannheim*

---

Twitter: [@c\\_janz](https://twitter.com/c_janz) | E-Mail: [christian.janz@bridging-it.de](mailto:christian.janz@bridging-it.de) | Slides: <http://de.slideshare.net/cjanz>  
| Code: <https://github.com/bridgingIT/angular-seed>

# Agenda

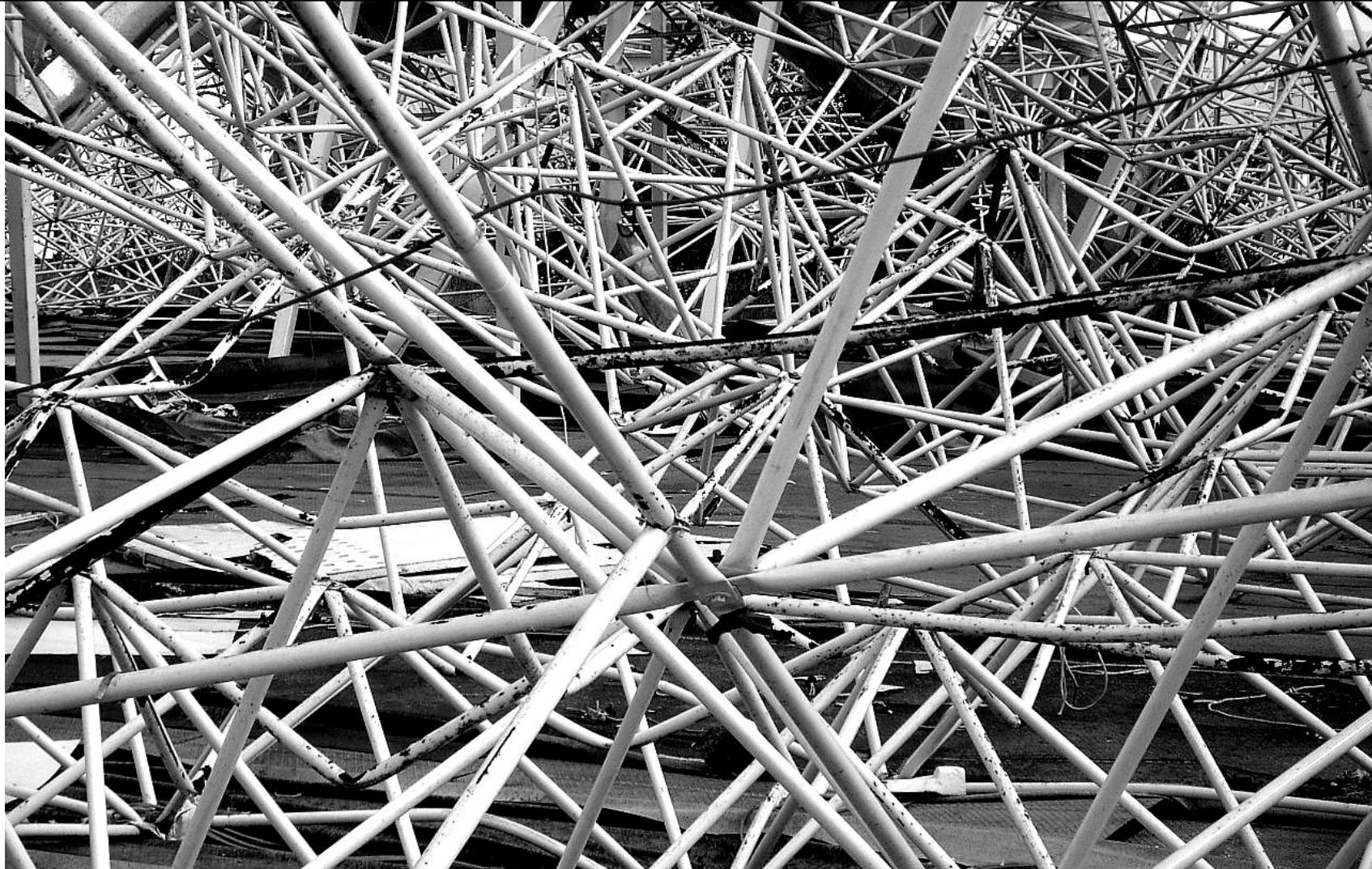
- Enterprise Applications
- Enterprise + JavaScript?
- AngularJS
- JavaScript Ökosystem
- Continuous Integration
- Fazit

# Enterprise Applications

Was ist besonders daran?

Enterprise Applications  
unterstützen kritische  
Geschäftsprozesse

# Hohe Komplexität



# Hohe Qualitätsanforderungen



# CHANGE

# Deadlines



Enterprise +  
JavaScript?

---

*“JavaScript ist doch für Animationen auf  
Webseiten. Damit kann man keine  
professionellen Geschäftsanwendungen  
entwickeln.”*

---

---

*“Der JavaScript-Code hat keine Unit-Tests, das  
geht nicht so einfach.”*

# Enterprise + JavaScript!



# AngularJS

---

*HTML enhanced for web apps!*

- JavaScript-Framework von Google
- Open Source (MIT)
- MVW (Model-View-Whatever)

# Features

- Module
- Dependency Injection
- Trennung von View und Logik
- Data Binding
- Erweiterung von HTML
- **Testability**

# Building Blocks

- Module
- View
- Route Definition
- Controller
- Service
- Directive
- Filter

# Demo

## InterpolateFilter + Unit Test

```
'use strict';

angular.module('myApp.version.interpolate-filter', [])

.filter('interpolate', ['$version', function(version) {
    return function(text) {
        return String(text).replace(/\%VERSION\%/mg, version);
    };
}]);
```

# JavaScript Ökosystem

# Testing



# Code Analysis



**JSLint**  
The JavaScript Code Quality Tool

# Package Management



# Bower: Beispiel

```
{  
  "name": "angular-seed",  
  "description": "A starter project for AngularJS",  
  "version": "0.0.0",  
  "homepage": "https://github.com/angular/angular-seed",  
  "license": "MIT",  
  "private": true,  
  "dependencies": {  
    "angular": "1.2.x",  
    "angular-route": "1.2.x",  
    "angular-loader": "1.2.x",  
    "angular-mocks": "~1.2.x",  
    "html5-boilerplate": "~4.3.0"  
  }  
}
```

# Task Runner



# Demo: Gulp

```
[22:53:53] Using gulpfile ./angular-seed/gulpfile.js
[22:53:53] Starting 'test'...
INFO [karma]: Karma v0.12.24 server started at http://localhost:9876/
INFO [launcher]: Starting browser Chrome
INFO [Chrome 38.0.2125 (Linux)]: Connected on socket P7CU4ehH9G-nJyq-796g with id :1
.....
Chrome 38.0.2125 (Linux): Executed 6 of 6 SUCCESS (0.052 secs / 0.049 secs)
[22:53:55] Finished 'test' after 2.27 s
[22:53:55] Starting 'build'...
./angular-seed/app/view2/view2.js: line 4, col 3, Missing "use strict" statement.

1 error

events.js:72
throw er; // Unhandled 'error' event
^
Error: JSHint failed for: ./angular-seed/app/view2/view2.js
```

# Continuous Integration

Bringing it all together

# Demo: Jenkins Job

**Buildverfahren**

Shell ausführen  
Befehl: `npm install`

[Liste der verfügbaren Umgebungsvariablen](#)

Shell ausführen  
Befehl: `gulp`

[Liste der verfügbaren Umgebungsvariablen](#)

Invoke Standalone Sonar Analysis  
Task to run:   
JDK: **(Inherit From Job)**  
JDK to be used for this sonar analysis  
Path to project properties:   
Project properties:   
JVM Options:

**Build-Schritt hinzufügen ▾**

**Post-Build-Aktionen**

Veröffentliche JUnit-Testergebnisse.

Testberichte in XML-Format: `test_out/*.xml`

Es sind reguläre Ausdrücke wie z.B. 'myproject/target/test-reports/\*.xml' erlaubt. Das genaue Format können Sie [der Spezifikation für @Includes eines Ant-Filesets](#) entnehmen. Das Ausgangsverzeichnis ist der [Arbeitsbereich](#).

Lange Standard-Out/-Error Ausgaben aufbewahren

# Demo: Code-Analyse mit SonarQube

Version 1.0.0 - 23. Okt 2014 09:37 [Time changes...](#)

|               |             |                    |
|---------------|-------------|--------------------|
| Lines Of Code | Files       | Functions          |
| 50            | 6           | 9                  |
| JavaScript    | Directories | Lines              |
|               | 4           | 70                 |
|               |             | Classes Statements |
|               |             | 0 19               |
|               |             | Accessors          |
|               |             | 0                  |

SQALE Rating A Technical Debt Ratio [9,3%](#)

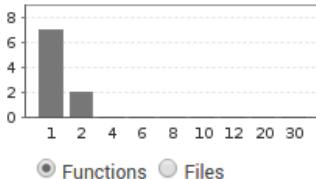
|                |        |          |   |
|----------------|--------|----------|---|
| Technical Debt | Issues | Blocker  | 0 |
| 2h 20min       | 7      | Critical | 0 |
|                |        | Major    | 1 |
|                |        | Minor    | 0 |
|                |        | Info     | 6 |

Unit Tests Coverage [52,4%](#)  
Line Coverage [52,4%](#)

Duplications [0,0%](#)

|       |        |       |
|-------|--------|-------|
| Lines | Blocks | Files |
| 0     | 0      | 0     |

Complexity [1,0 /function](#)  
[1,5 /file](#)  
Total: 9



# Fazit

Mit AngularJS können  
professionelle  
Geschäftsanwendungen  
entwickelt werden

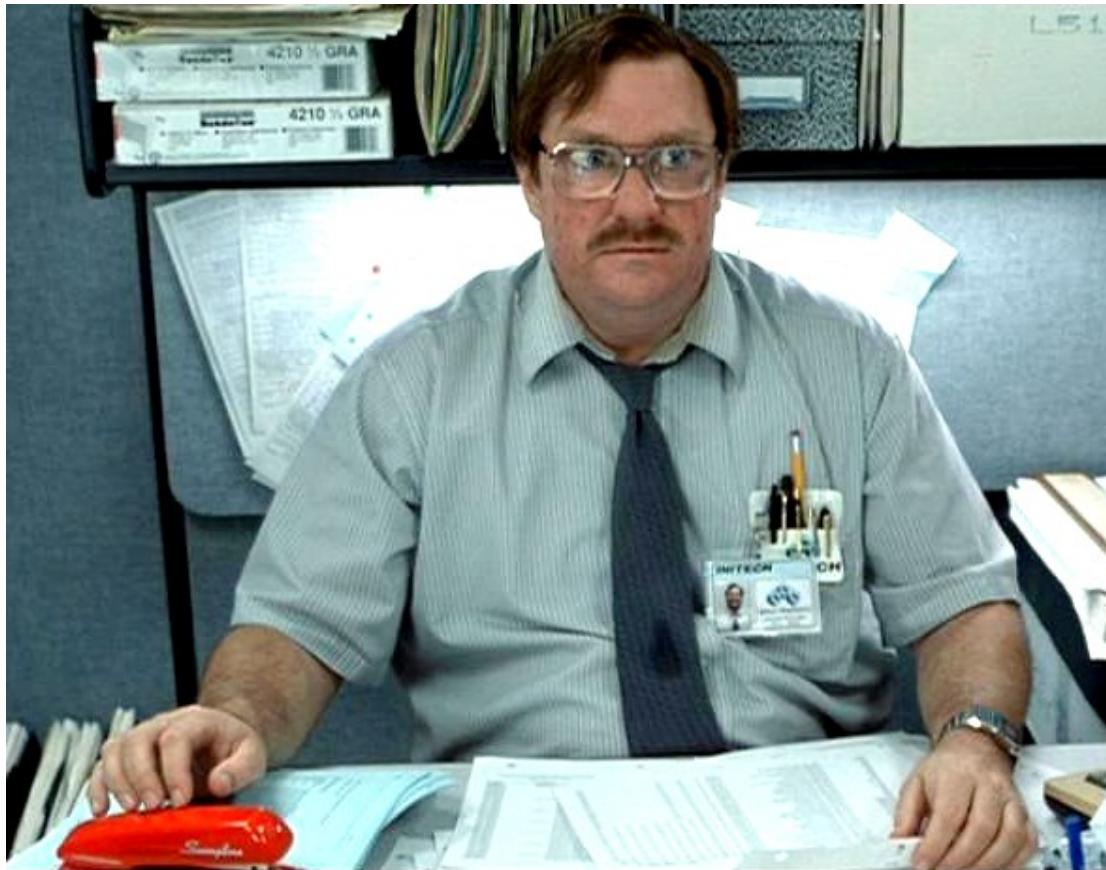
# Es gibt sehr gute Werkzeuge



# Bewährte Methoden lassen sich übertragen



# Fragen?



# Bildnachweise

- <https://www.flickr.com/photos/dominik99/384027019/>
- <http://cdn.playbuzz.com/cdn/8ba53e3b-5c1b-4fe0-8d93-f9586b7f4a97/73917b9f-1ced-41fd-8e97-986c8754a01c.jpg>
- [http://commons.wikimedia.org/wiki/File:Neon\\_sign,\\_%22CHANGE%22\\_in\\_Broadway\\_Neon\\_Sign\\_in\\_NYC.jpg](http://commons.wikimedia.org/wiki/File:Neon_sign,_%22CHANGE%22_in_Broadway_Neon_Sign_in_NYC.jpg)
- [https://www.blossom.io/static/img/site/product-mgmt-anti-patterns/office\\_space.jpg](https://www.blossom.io/static/img/site/product-mgmt-anti-patterns/office_space.jpg)
- [http://commons.wikimedia.org/wiki/File:Space\\_Shuttle\\_Discovery\\_%28Orbiter\\_10%29\\_at\\_Kennedy\\_Space\\_Center\\_in\\_2009.jpg](http://commons.wikimedia.org/wiki/File:Space_Shuttle_Discovery_%28Orbiter_10%29_at_Kennedy_Space_Center_in_2009.jpg)
- <http://blog.shoeboxed.com/4-lessons-we-can-learn-from-the-movie-space/>