



# Please plug in the TDD

---

STIAN GALAPATE-EDVARDBSEN

[HTTPS://WWW.LINKEDIN.COM/IN/STIAN-GALAPATE-EDVARDBSEN-A678857/](https://www.linkedin.com/in/stian-galapate-edvardsen-a678857/)

# The challenge

---

- The leap from the TDD basics and your everyday reality can be huge
- You need to learn to walk before you can run
- But you are already running (although perhaps using one foot and one arm)
- And you can't stop

# The challenge – a sample

---

- Your project is a rewrite and enhancement of a legacy system of a massive size
- Ye olde dinosaur shall now live in the cloud and scale indefinitely
- More importantly, it should have a bunch of new features to justify the investment
- (no, living in the cloud and indefinite scaling is not a feature. Those shiny new auto-reports are)

# The way

---

- You now know TDD
- Tests are created. They are red. Then they are green.
- Your codebase doesn't have smells, your codebase is one big blue cheese
- You refactor aggressively

# Intermission

## COHORT 3 REPO

```
github.com/AlDanial/cloc v 1.90 T=0.72 s (39.0 files/s, 8148.2 lines/s)
```

Language	files	blank	comment	code
JSON	4	0	0	4986
C#	15	119	57	524
XML	4	0	0	63
Visual Studio Solution	1	1	0	21
MSBuild script	2	7	0	19
C# Generated	2	8	20	18
SUM:	28	135	77	5631

```
C:\repos\work\bouvet-cohort-03>
```

## SAMPLE REPO

```
github.com/AlDanial/cloc v 1.90 T=3552.91 s (15.6 files/s, 4237.7 lines/s)
```

Language	files	blank	comment	code
JavaScript	29448	641554	936212	5028302
XML	1954	15702	19954	4231700
JSON	8392	432	0	1790991
Markdown	3301	161300	4	389927
TypeScript	6019	36036	208095	325359
CSV	20	0	0	299147
C#	3614	56427	38546	291756
HTML	270	20382	3108	147489
CSS	213	8485	2365	78590
SQL	30	243	166	35998
MSBuild script	296	570	1267	35805
C++	69	3720	3470	22045
Python	54	4204	7644	20083
Visual Studio Solution	67	58	66	19948
C/C++ Header	102	3189	1953	17491
Sass	273	3633	2913	16788
C# Designer	141	1787	5320	16537
Java	168	3126	8365	13964
Web Services Description	26	0	0	10562
YAML	291	1095	917	8748
XSD	33	51	6	7880
LESS	55	846	985	5854
Razor	135	1725	164	5525
SVG	7	0	1	4765
PowerShell	106	710	2927	3694
Bourne Shell	98	457	329	2617
ASP.NET	75	150	58	2507
DOS Batch	97	185	3	1350
C# Generated	113	450	1122	1070
CoffeeScript	19	396	47	1005
XSLT	4	39	18	827
Windows Module Definition	7	140	0	656
make	17	177	38	593
diff	1	3	13	588
Bourne Again Shell	14	83	22	359
Lisp	3	42	38	264
m4	2	39	2	260
PHP	2	42	19	231
Puppet	1	32	0	223
XAML	2	26	3	179
C	2	24	27	111
EJS	2	4	0	92
INI	3	0	0	84
Perl	1	16	30	57
JSON5	1	0	0	55
TOML	1	2	0	39
Ant	1	10	0	33
Windows Resource File	1	1	1	33
Visual Basic .NET	3	8	17	26
Handlebars	1	5	0	23
Nix	1	1	0	19
F#	1	0	1	4
Ruby	1	0	2	4
ASP	1	0	0	1
SUM:	55559	967607	1246238	12842258

# The way - continued

---

- Your tests are still green
- But nothing works anymore except the functionality that your two tests test
- You write more tests so that more things can work
- You refactor aggressively

# The way - continued

---

- Nothing compiles anymore and you don't know why
- The customers wonder about the new auto-report and if they can have it before lunch the next day

# The journey is the destination

---

- We've learned wishful thinking, and making the least-effort to fulfill the wishful thinking
- We can now use the transformation priority premise and object calisthenics to keep our code sleek and maintainable
- We test on behavior and not implementation as to enable changes to implementation details without getting clogged in hundreds of (when all comes to all) irrelevant tests
- And we can sort of do it as a group
- We still have a ways to go



# Thank you!

---

And looking forward to the next module 😊