

**ACCU
2021**
VIRTUAL EVENT

Bloomberg
Engineering

undo

 **mosaic**
CONSULTANTS TO FINANCIAL SERVICES

Tools That Spark Joy: Lessons Learned From the Rust Ecosystem That Can Be Adopted Elsewhere

Lotte Steenbrink



TOOLS THAT SPARK JOY

lessons learned from the Rust ecosystem that can be adopted elsewhere

LOTTE STEENBRINK, FERROUS SYSTEMS

Hello!

LOTTE STEENBRINK, SOFTWARE ENGINEER

- IoT, constrained networks, embedded systems
- IETF
- C & C++

AT FERROUS SYSTEMS



- Build Trust
- Reduce Friction and Fear
- Spark Joy

OUTLINE

- 1. built-in Rust tools**
- 2. 3rd party open source tooling**
- 3. lessons learned**

Rust is a systems programming language that is optimised for speed, memory safety and fearless concurrency.

Rust is a systems programming language that is optimised for speed, memory safety and fearless concurrency.

**RUST IS AN EMPATHIC
SYSTEMS PROGRAMMING
LANGUAGE THAT IS
DETERMINED TO NOT LET
ME SHOOT MYSELF IN THE
FOOT.**

let me show you what I mean.

```
fn main() {  
    let answer = 0;  
    // do some more calculations ...  
    answer = 42;  
    println!("The Answer to the \  
Ultimate Question of Life, \  
the Universe, and Everything \  
is: {}", answer);  
}
```

let me show you what I mean.

```
fn main() {  
    let answer = 0;  
    // do some more calculations ...  
    answer = 42;  
    println!("The Answer to the \  
Ultimate Question of Life, \  
the Universe, and Everything \  
is: {}", answer);  
}
```

```
error[E0384]: cannot assign twice to immutable variable `answer`  
  --> src/bin/deep_thought.rs:10:5  
    |  
8   | let answer = 0;  
    |     -----  
    |     |  
    |     first assignment to `answer`  
    |     help: make this binding mutable: `mut answer`  
9   | // do some more calculations ...  
10  | answer = 42;  
    |     ^^^^^^^^^^^^^ cannot assign twice to immutable variable
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384``.

talk to me like i'm a person

```
error[E0384]: cannot assign twice to immutable variable `answer`
--> src/bin/deep_thought.rs:10:5
   |
8  | let answer = 0;
   |     -----
   |     |
   |     first assignment to `answer`
   |     help: make this binding mutable: `mut answer`
9  | // do some more calculations ...
10 | answer = 42;
   | ^^^^^^^^^^^ cannot assign twice to immutable variable
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384`.

```
error[E0384]: cannot assign twice to immutable variable `answer`
--> src/bin/deep_thought.rs:10:5
   |
8  | let answer = 0;
   |     -----
   |     |
   |     first assignment to `answer`
   |     help: make this binding mutable: `mut answer`
9  | // do some more calculations ...
10 | answer = 42;
   | ^^^^^^^^^^^ cannot assign twice to immutable variable
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384`.

show me the
problem

```
error[E0384]: cannot assign twice to immutable variable `answer`
```

```
--> src/bin/deep_thought.rs:10:5
```

```
8 | let answer = 0;
```

```
    |-----
```

```
    | first assignment to `answer`
```

```
    | help: make this binding mutable: `mut answer`
```

```
9 | // do some more calculations ...
```

```
10 | answer = 42;
```

```
    | ^^^^^^^^^^^^ cannot assign twice to immutable variable
```

```
error: aborting due to previous error
```

```
For more information about this error, try `rustc --explain E0384`.
```

emphasise
carefully



```
error[E0384]: cannot assign twice to immutable variable `answer`
--> src/bin/deep_thought.rs:10:5
   |
8  | let answer = 0;
   |     -----
   |     |
   |     first assignment to `answer`
   |     help: make this binding mutable: `mut answer`
9  | // do some more calculations ...
10 | answer = 42;
   | ^^^^^^^^^^^ cannot assign twice to immutable variable
```

suggest a solution

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384`.

group by
whitespace

```
error[E0384]: cannot assign twice to immutable variable `answer`
```

```
-> src/bin/deep_thought.rs:10:5
```

```
8 | let answer = 0;
```

```
    -----
```

```
    |  
    | first assignment to `answer`
```

```
    | help: make this binding mutable: `mut answer`
```

```
9 | // do some more calculations ...
```

```
10 | answer = 42;
```

```
    | ^^^^^^^^^^^ cannot assign twice to immutable variable
```

```
error: aborting due to previous error
```

```
For more information about this error, try `rustc --explain E0384`.
```

```
error[E0384]: cannot assign twice to immutable variable `answer`
--> src/bin/deep_thought.rs:10:5
   |
8  | let answer = 0;
   |     -----
   |     |
   |     first assignment to `answer`
   |     help: make this binding mutable: `mut answer`
9  | // do some more calculations ...
10 | answer = 42;
   | ^^^^^^^^^^^^ cannot assign twice to immutable variable
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384`.

delete noise

started from the bottom, now we're here

```
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:29:22: 29:30 error: cannot borrow `foo.bar1` as mutable more than once at a time [E0499]
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:29     let _bar2 = &mut foo.bar1;
                                   ~~~~~
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:29:22: 29:30 help: run `rustc --explain E0499` to see a detailed explanation
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:28:21: 28:29 note: previous borrow of `foo.bar1` occurs here; the mutable borrow prevents
subsequent moves, borrows, or modification of `foo.bar1` until the borrow ends
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:28     let bar1 = &mut foo.bar1;
                                   ~~~~~
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:31:2: 31:2 note: previous borrow ends here
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:26 fn borrow_same_field_twice_mut_mut() {
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:27     let mut foo = make_foo();
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:28     let bar1 = &mut foo.bar1;
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:29     let _bar2 = &mut foo.bar1;
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:30     *bar1;
src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:31 }
```

error[E0384]: cannot assign twice to immutable variable `answer`

```
--> src/bin/deep_thought.rs:10:5
  |
  |
8 | let answer = 0;
  |     -----
  |     |
  |     first assignment to `answer`
  |     help: make this binding mutable: `mut answer`
9 | // do some more calculations ...
10| answer = 42;
   | ^^^^^^^^^^^ cannot assign twice to immutable variable
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0384`.

ready, set, rust 🏁

```
→ ferrous cargo new
```


ready, set, rust



```
→ ferrous cargo new --lib xoxo  
   Created library `xoxo` package
```

```
→ ferrous cd xoxo
```

```
→ xoxo git:(master) x tree
```

```
.  
├── Cargo.toml  
└── src  
    └── lib.rs
```

```
1 directory, 2 files
```

```
→ xoxo git:(master) x █
```

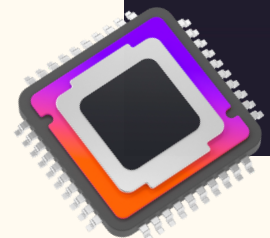
when the grass is greener on the other side, accept their gardening tips

```
bin / . . . . . / main
#![no_main]
#![no_std]
use my_app as _;

▶ Run | Debug
#[cortex_m_rt::entry]
fn main() -> ! {
    ... defmt::warn!("Hello, ACCU!");
    ... defmt::error!("👋😄");
    💡 defmt::info!["How are you?"];

    ... my_app::exit()
}
```

```
→ my-app git:(main) x cargo install cargo-watch
   Updating crates.io index
   Ignored package `cargo-watch v7.5.1` is already installed, use --force to
→ my-app git:(main) x cargo watch -x '█'
```



not invented here 👍

```
[profile.dev]
```

```
debug = 1
```

Cargo.toml

```
[target.'cfg(all(target_arch = "arm", target_os = "none"))']
```

```
runner = "probe-run --chip nRF52840_xxAA"
```

./cargo.config

→ ~ cargo install probe-run

```
bin / hello.rs / main
#![no_main]
#![no_std]
use my_app as _;

▶ Run | Debug
#[cortex_m_rt::entry]
fn main() -> ! {
    ... defmt::warn!("Hello, ACCU!");
    ... defmt::error!("👋😭");
    ... defmt::info!("How are you?");
    ... my_app::exit()
}

(HOST) 0 success!
WARN Hello, ACCU!
└─ hello::_cortex_m_rt_main @ src/bin/hello.rs:7
ERROR 👋😭
└─ hello::_cortex_m_rt_main @ src/bin/hello.rs:8
INFO How are you?
└─ hello::_cortex_m_rt_main @ src/bin/hello.rs:9
stack backtrace:
0: lib::inline::_bkpt
   at ./asm/inline.rs:13
1: __bkpt
   at ./asm/lib.rs:49
2: my_app::exit
   at src/lib.rs:18
3: hello::_cortex_m_rt_main
   at src/bin/hello.rs:11
4: main
   at src/bin/hello.rs:5
5: ResetTrampoline
   at /Users/lottesteenbrink/.cargo/registry/src/github.com-1ecc6299db9e
   .6.13/src/lib.rs:547
6: Reset
   at /Users/lottesteenbrink/.cargo/registry/src/github.com-1ecc6299db9e
   .6.13/src/lib.rs:550
[Finished running. Exit status: 0]
```

Make the right thing to do easy

– Ashley Williams

```
uite > tests > test.rs > {} tests > assert_eq
```

```
#![no_std]
#![no_main]
```

```
use my_app as _;
```

▶ Run Test | Debug

```
#[defmt_test::tests]
```

```
mod tests {
```

```
    use defmt::assert_eq;
```

▶ Run Test | Debug

```
    #[test]
```

```
    fn assert_eq() {
```

```
        assert_eq!(24, 42, "The answer is wrong")
```

```
    }
```

```
}
```

```
> Executing task: cargo test --package testsuite --test test -- tests --nocapture <
```

```
Finished test [optimized + debuginfo] target(s) in 0.12s
```

```
Running target/thumbv7em-none-eabihf/debug/deps/test-d06727ad21b3eb46
```

```
(HOST) INFO flashing program (7.50 KiB)
```

```
(HOST) INFO success!
```

```
INFO (1/1) running `assert_eq`...
```

```
└─ test::tests::__defmt_test_entry @ tests/test.rs:6
```

```
ERROR panicked at 'assertion failed: `(left == right)`: The answer is wrong'
```

```
diff < left / right >
```

```
<24
```

```
>42
```

```
└─ test::tests::assert_eq @ tests/test.rs:12
```

```
stack backtrace:
```

```
0: HardFaultTrampoline
```

```
<exception entry>
```

```
1: lib::inline::__udf
```

```
   at ./asm/inline.rs:171
```

```
2: __udf
```

```
   at ./asm/lib.rs:49
```

```
3: cortex_m::asm::udf
```

```
   at /Users/lottesteenbrink/.cargo/registry/src/github.com-1ecc6299db9ec823/cortex
```


make the right thing to do *enticing*

```
bin / levels.rs / main
#[derive(Format)]
struct Person {
    name: Str,
    age: u8,
}

▶ Run | Debug
#[cortex_m_rt::entry]
fn main() -> ! {
    let interned_name: Str = defmt::intern!("Lotte");

    defmt::info!("Hello 🙌 😊");
    defmt::warn!("{=?}", Person {
        name: interned_name,
        age: 30
    });
    defmt::error!("{0=0..3:x}", 0b0100_0111u8);

    my_app::exit()
}

> Executing task: cargo run --package my-app --bin levels <

Compiling my-app v0.1.0 (/Users/lottesteinbrink/ferrous/my-app)
Finished dev [optimized + debuginfo] target(s) in 0.56s
Running `probe-run --chip nRF52840_xxAA target/thumbv7em-no
vels`
(HOST) INFO flashing program (5.98 KiB)
(HOST) INFO success!

INFO Hello 🙌 😊
└─ levels::__cortex_m_rt_main @ src/bin/levels.rs:17
    WARN Person { name: Lotte, age: 30 }
└─ levels::__cortex_m_rt_main @ src/bin/levels.rs:18
    ERROR 0x7
└─ levels::__cortex_m_rt_main @ src/bin/levels.rs:22
stack backtrace:
0: lib::inline::__bkpt
   at ./asm/inline.rs:13
1: __bkpt
   at ./asm/lib.rs:49
2: my_app::exit
   at src/lib.rs:18
3: levels::__cortex_m_rt_main
   at src/bin/levels.rs:24
4: main
   at src/bin/levels.rs:13
5: ResetTrampoline
```

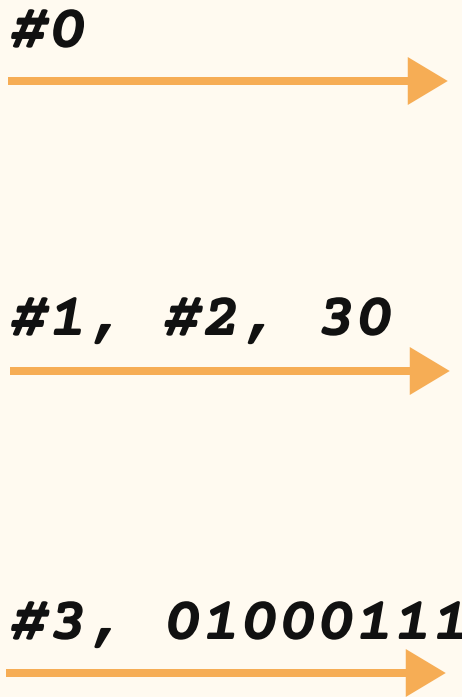
make the right thing to do *enticing*



```
defmt::info!("Hello 🙌 😊");

defmt::warn!("{=?}",
  Person { name: interned_name,
            age: 30 } );

defmt::error!("{0=0..3:x}",
  0b0100_0111u8);
```

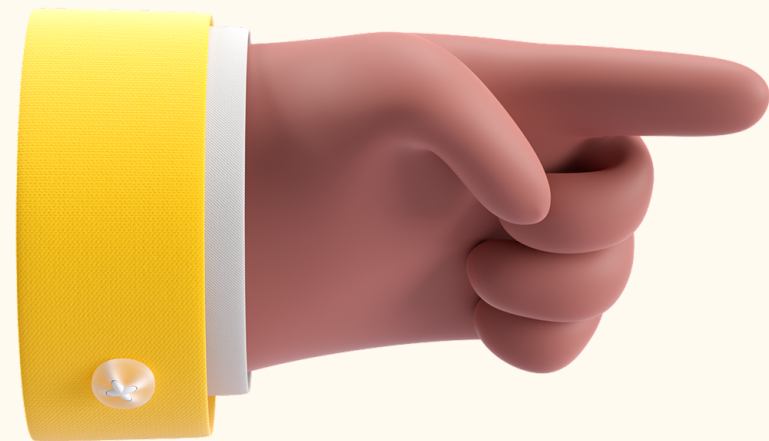


#0	"Hello 🙌 😊"	🌸
#1	"{=?}"	👉
#2	"Lotte"	
#3	"{0=0..3:x}"	!

```
INFO Hello 🙌 😊

WARN { name: Lotte, age: 30 }


ERROR 0x7
```



LESSONS LEARNED, THE LISTICLE™

PRESENTATION IS KEY

- delete noise
- whitespace groups information
- *carefully* emphasize



```
error[E0384]: cannot assign twice to immutable variable `answer`
--> src/bin/hello.rs:10:5
 8 |     let answer = 0;
   |     -----
   |         |
   |         first assignment to `answer`
   |         help: make this binding mutable:
 9 |         // do some more calculations ...
10 |         answer = 42;
   |         ^^^^^^^^^^^ cannot assign twice to immutable variable `answer`

error: aborting due to previous error; 1 warning emitted

For more information about this error, try `rustc --explain E0384`.
```


PUT HELP WHERE YOUR AUDIENCE NEEDS IT

- reduce context switches
- show and tell
- ♥ copy-paste-able solutions



IF YOU HAD TO SUFFER THROUGH IT, MAKE SURE NOBODY ELSE HAS TO

- Explain it to the human, not their computer
- Automate setup steps



LEAD BY EXAMPLE

- set a standard to live up to and copy from
- *if possible*, consider ✨joy✨ at the design stage
- sponsor open source tools that make your life easier

STEAL GOOD IDEAS

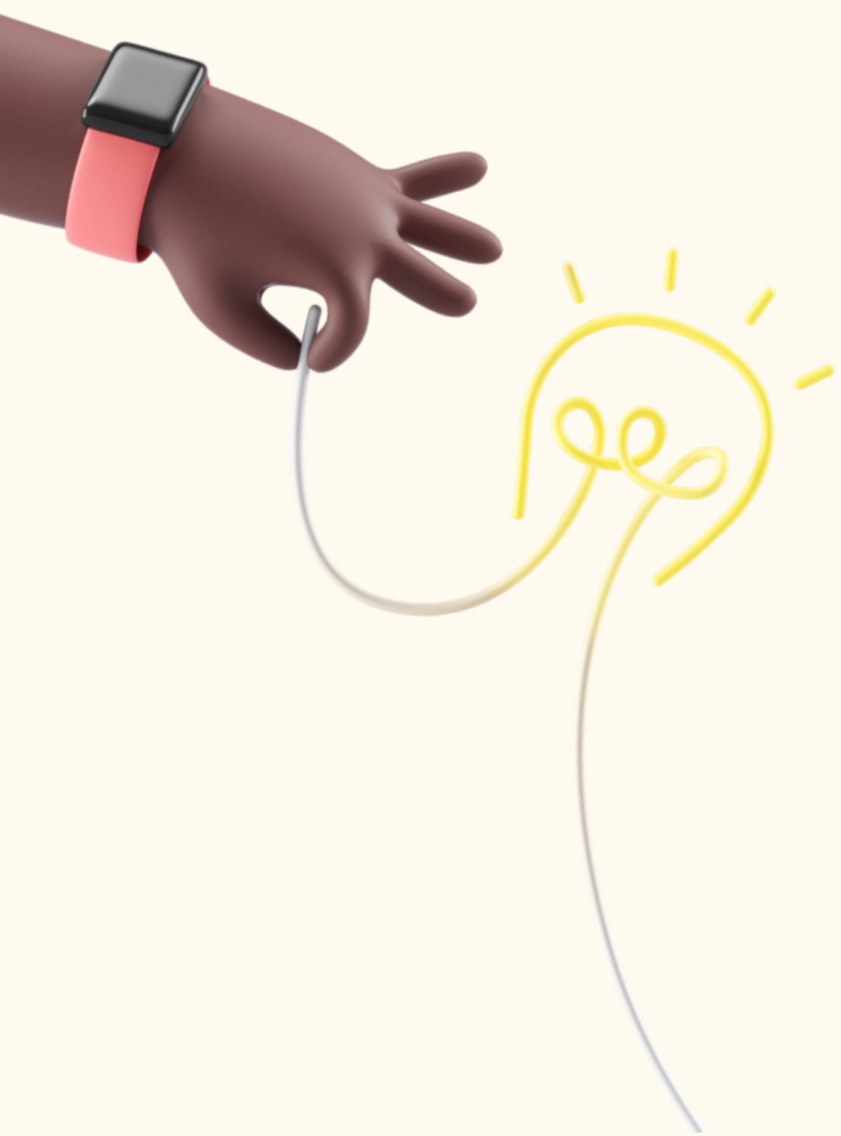
**KEEP THE HAPPY PATH
CONSISTENT**

MAKE THE RIGHT THING TO DO EASY

– Ashley Williams

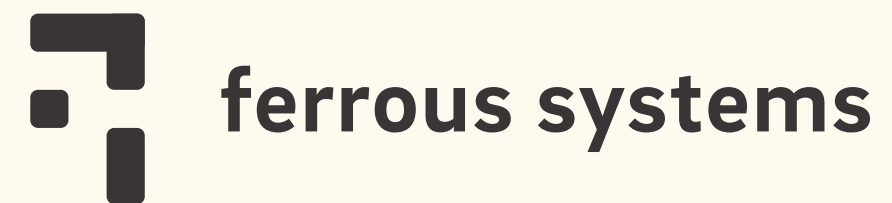


**MAKE THE RIGHT
THING TO DO
*ENTICING***



✉ lotte.steenbrink@ferrous-systems.com

🐙 github.com/Lotterleben



SLIDES, SOURCES & MATERIALS

🔗 tinyurl.com/accu-spark

DEAR INTERVIEWEES: THANK YOU

Florian Gilcher, Jorge Aparicio, James Munns,

Jan-Erik Rediger

Sources

[🔗](#) click on each link to get to the source!

TOOLS

[knurling-rs](#) (sponsor us!)

[defmt](#) | [postform, a defmt-inspired C++ library.](#)

[cargo-watch](#) | [entr](#)

▶ Run button and *much* more IDE magic by [rust-analyzer](#)

[Rust Playground](#) | [C Playground](#)

TALKS

[Bending the Curve: A Personal Tutor at Your Fingertips](#)

[The Anatomy of Error Messages in Rust](#)

[C++ & Rust: Access all Arenas](#)

[Zen and the Art of Convincing Your Company to Use Rust](#)

[RustConf 2020 Keynote](#)

[A Tale of Two Asyncs: Open Source Language Design in Rust and Node.js](#)

Sources

[🔗](#) *click on each link to get to the source!*

TEXT

elm-lang.org/news/compiler-errors-for-humans

blog.rust-lang.org/2016/08/10/Shape-of-errors-to-come.html

[usability improvements in gcc 9](#)

[gcc User Experience Guidelines](#)

[rustc diagnostics style guide](#)

[extending cargo with custom commands](#) – 1 paragraph

[Testing a Hardware Abstraction Layer \(HAL\) using defmt](#)

OTHER

3D illustrations by [craftwork.design](#)

Chubby hands by [pitch.com](#)



TOOLS THAT SPARK JOY

lessons learned from the Rust ecosystem that can be adopted elsewhere

LOTTE STEENBRINK, FERROUS SYSTEMS

```
12
13 [dependencies]
14 addr2line = "0.14.1" ✓
15 ansi_term = "0.12.1" ✓
16 anyhow = "1.0.32" ✓
17 arrayref = "0.3.6" ✓
18 colored = "2.0.0" ✓
19 defmt-decoder = { git = "https://github.com/knurling-rs/defmt", tag = "defmt-decoder-v0.2.0",
20 difference = "2.0.0" ✓
21 gimli = "0.23.0" ✓
22 log = "0.4.11" ✓
23 # an addr2line trait is implement for a type in this particular version
24 object = "0.22.0" ✗ 0.23.0
25 probe-rs = "0.10.0" ✓
26 probe-rs-rtt = "0.10.0" ✓
27 rustc-demangle = "0.1.16" ✓
28 signal-hook = "0.3.4" ✓
29 structopt = "0.3.15" ✓
30 hidapi = "1.2.5" ✓
31
```



Esteban K – 🦀🔧@home @ekuber · Dec 6, 2020

Just out of curiosity, was the error not clear for some reason? Could it be made easier to understand?

play.rust-lang.org/?version=stabl...

```
error: expected `;`, found `}`
```

```
--> src/main.rs:5:6
```

```
5 |         }  
  |         ^ help: add `;` here  
6 |     }  
  |     - unexpected token
```



1



Jakub T. Jankiewicz (λ)

@jcubic

Replying to @ekuber and @IlijaMarijanov8

I'm not used to good error messages, so I didn't look at it actually LOL

8:54 AM · Dec 7, 2020 · Twitter Web App

2 Likes

beware of survivorship bias

 **Esteban K** – 🦀 ⚙️ @home
@ekuber

"No need to address that, no-one is complaining about it."



1:19 AM · Feb 25, 2021 · Twitter for Android

21 Retweets 1 Quote Tweet 134 Likes