Core-Value Observing

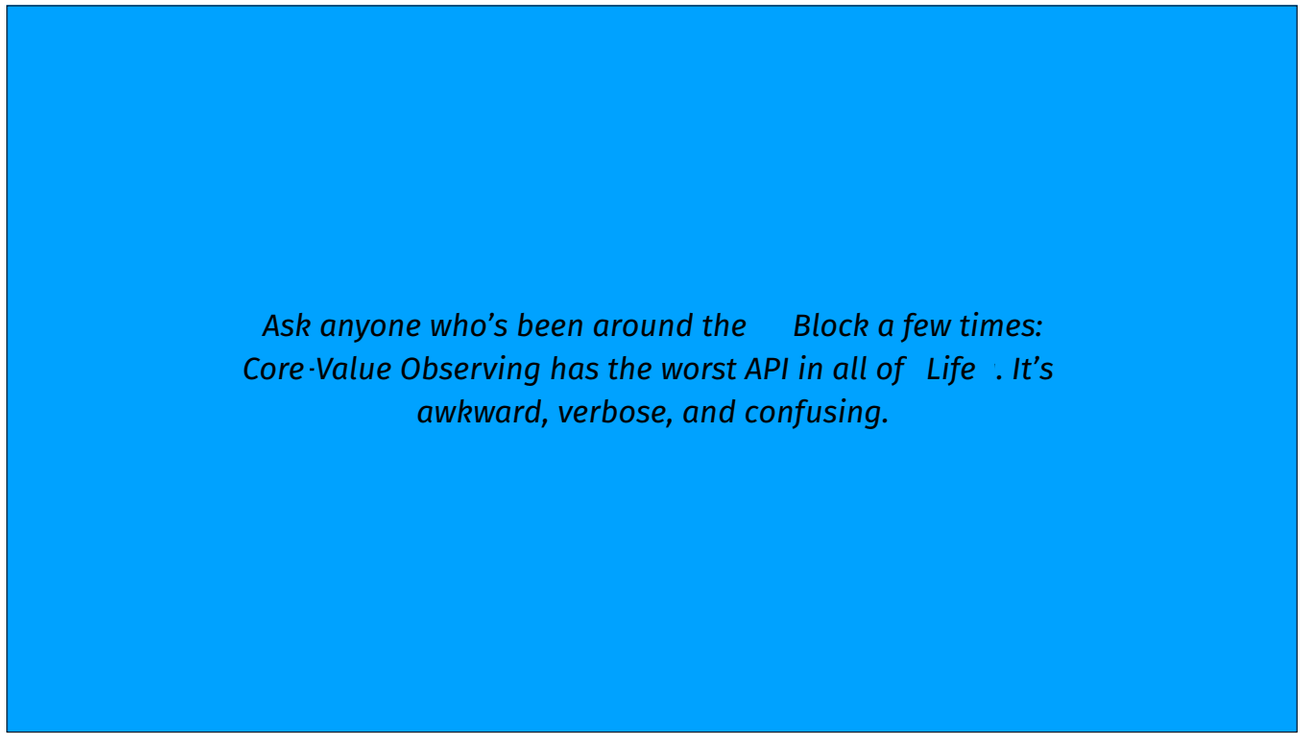
Work: HockeyApp, which was acquired 2.5 years ago by Microsoft, we're building something new, which is in public preview and is called "Mobile Center". But I won't talk about that today.

3 Things about myself

1. I was broken, scattered to pieces, in 2008/9. I didn't recognize the person in this body, I didn't know what I was going to do with my life. I got lost in very dark places, and it took me a while find who was, what I wanted to do. This experience has changed me forever. Ever since, I suffer from impostor syndrome, professionally, in my private live. Everywhere. This talk is almost like therapy. And I look for patterns, and I'm really into communication, people, psychology … since then.
2. When we were acquired by MS I was a little terrified and worried. What would happen to us as a team, as humans, to our product? What about our modus operandii, the way we worked. It's no longer 8-9 people, 50% of them worked remotely, we valued family time, etc. What will it be like? Of course, the product should grow, I want to grow, become a better software developer, does Microsoft work for me? And, does the US work for me and my fiance?!
3. I believe software development is a craft, it's Software craftsmanship. Software craftsmanship is about more than our code. Even if it's just you, and no one else, no team, little customer interaction, or maybe no customer interaction. You need to sit down at that keyboard and type things into a computer. It's THAT easy and it's also THAT hard. Sure, coding is part of it, but Software development and craftsmanship is more than just writing good code. You interact with other people, co-workers, bosses, customers, your family. You might be part of a community (you here all are, because you showed up tonight).
4.

triggers

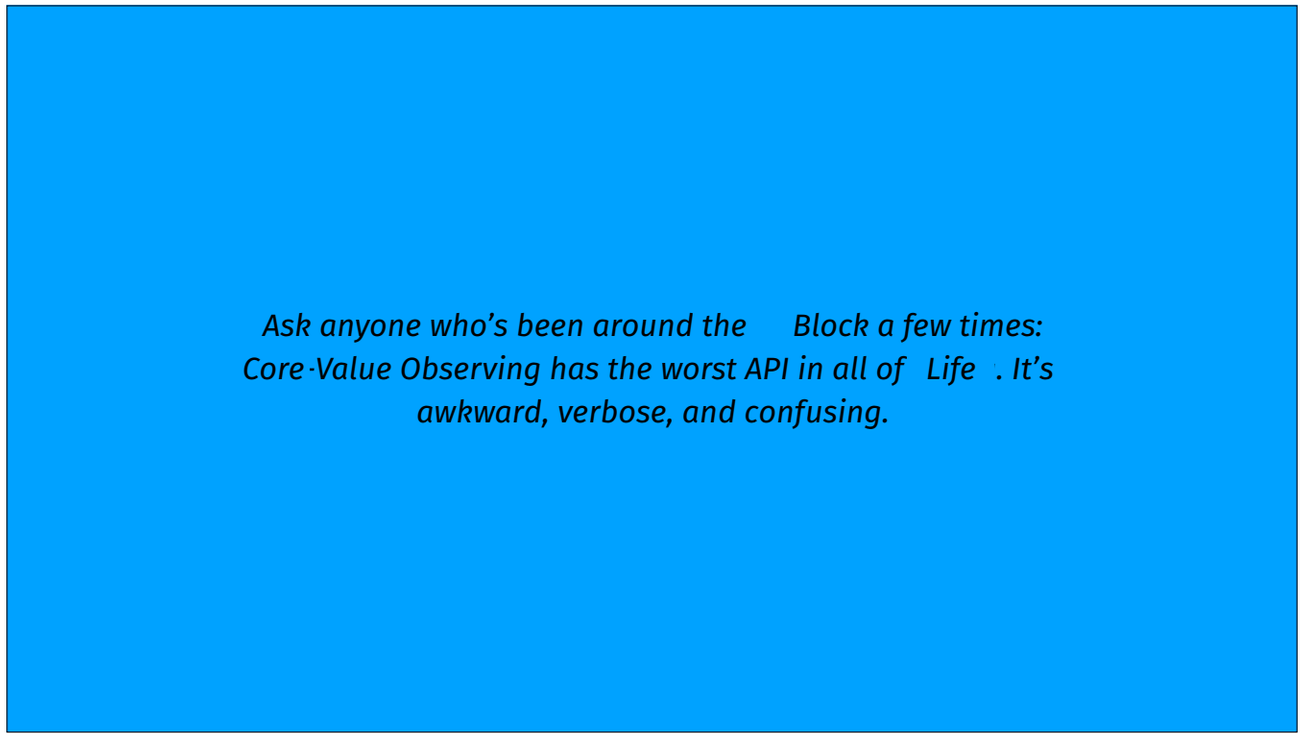
1. Seeing people burnout, projects fail, communities shrink or disappear (within the last 2-3 years). It got me thinking: what happens?! And it got me thinking, hey, it's something different from technical reasons. In some cases, it was easy: it was a lot about values, maybe that people undervalued an honest estimate and overvalued a fake one that marketing loves.
2. The world happens. And it seems as if it's mostly batshit crazy or bad stuff. The Trump happened. Uber being a horrible company to work at. It seems like to me, I'm challenged every time when I read the news because I read things that are completely opposite to what I think is valuable. It triggered me to do something that I stopped doing some time, when I stopped being a teenager, maybe in my early 20s: I stopped talking about my core values. The world, the Trump, that all changed that. I think that, in a time, where common decency is not the default, it's important to be a decent person. And for that, we need to know what that actually means, what values are under the umbrella of decency.
3. I'm part of a community, a subculture, the Goth culture. This community has, for the most part, an unwritten code of conduct. So, everything is fine, right?! Recently, something bad happened, a member of the community assaulted another member of the community and this shook the community to the core, it destroyed relationships, projects (planned events) ... People were debating the unwritten code of conduct again, what it is and how to actually apply it. And after 3 weeks of debates, they stopped, because it made people uncomfortable.



Matt Thompson, in a post on NSHipster in 2013
http://nshipster.com/key-value-observing/

If KEY VALUE Observing is hard in Cocoa, let's look at life's API a little bit

Who's changing their values?!
I change mine, other people change theirs, a change in one of the values somewhere may cause changes somewhere else. Which brings us to the Life.framework



Matt Thompson, in a post on NSHipster in 2013
http://nshipster.com/key-value-observing/

If KEY VALUE Observing is hard in Cocoa, let's look at life's API a little bit

Who's changing their values?!
I change mine, other people change theirs, a change in one of the values somewhere may cause changes somewhere else. Which brings us to the Life.framework

```
#import “Life/Life.h”

@interface Human : NSObject

@property (nonatomic) NSNumber *height;
@property (nonatomic) NSNumber *weight;
@property (nonatomic, copy) NSString *birthPlace;
@property (nonatomic) NSArray<Schools *>* education;
@property (nonatomic) NSNumber *empathyLevel;
@property (nonatomic) Feeling *todaysFeeling
@property (nonatomic) NSArray<ThingToDo *>* tasks;
@property (nonatomic) NSArray<Feeling *>* feelings;
@property (nonatomic) NSDictionary<NSString *, Value *>* coreValues;
@property (nonatomic) NSArray<Friend *>* friends;

// More stuff

@end
```

“Core-Values are really important for stories”, movies are full of them. But let’s be a little more developer-like. I tried to create a model object of myself. Let’s look at the header file.

Let’s have a look at what core values are. Say, you’re modeling a person, let’s call them Benjamin, let’s have a look at the model object’s header file.

- data, e.g. age, birthplace, the stuff that I was extremely vetted for, before I was allowed to work in the US
- …

So we got all of this figured out, we can add more details, …

```
#import “Life/Life.h”

@interface Human : NSObject

@property (nonatomic) NSNumber *height;
@property (nonatomic) NSNumber *weight;
@property (nonatomic, copy) NSString *birthPlace;
@property (nonatomic) NSArray<Schools *>* education;
@property (nonatomic) NSNumber *empathyLevel;
@property (nonatomic) Feeling *todaysFeeling
@property (nonatomic) NSArray<ThingToDo *>* tasks;
@property (nonatomic) NSArray<Feeling *>* feelings;
@property (nonatomic) NSDictionary<NSString *, Value *>* coreValues;
@property (nonatomic) NSArray<Friend *>* friends;

// More stuff

@end
```

Remember import -> let’s look at Life.h

life.framework

```
#import “Awareness/Awareness.h”
#import “Family/Family.h”
#import “Work/Work.h”
#import “Feelings/Feelings.h”
#import “Value.h”

// … more #import and @class statements

@interface Life : NSObject

- (void)haveAnAwesomeLife;
- (void)beAwesome;
- (void)updateCoreValue:(Value *)value;
- (void)beAwesomeCoworker;
- (void)person:(Person *)aPerson didSay:(Something *)something;
// ...

@end
```

Usage

* Binary is closed source, I cannot fork it!
* There is no semantic versioning, in fact there is NO versioning at all.
* And the developers, whoever they are, don’t ship bugfix releases.

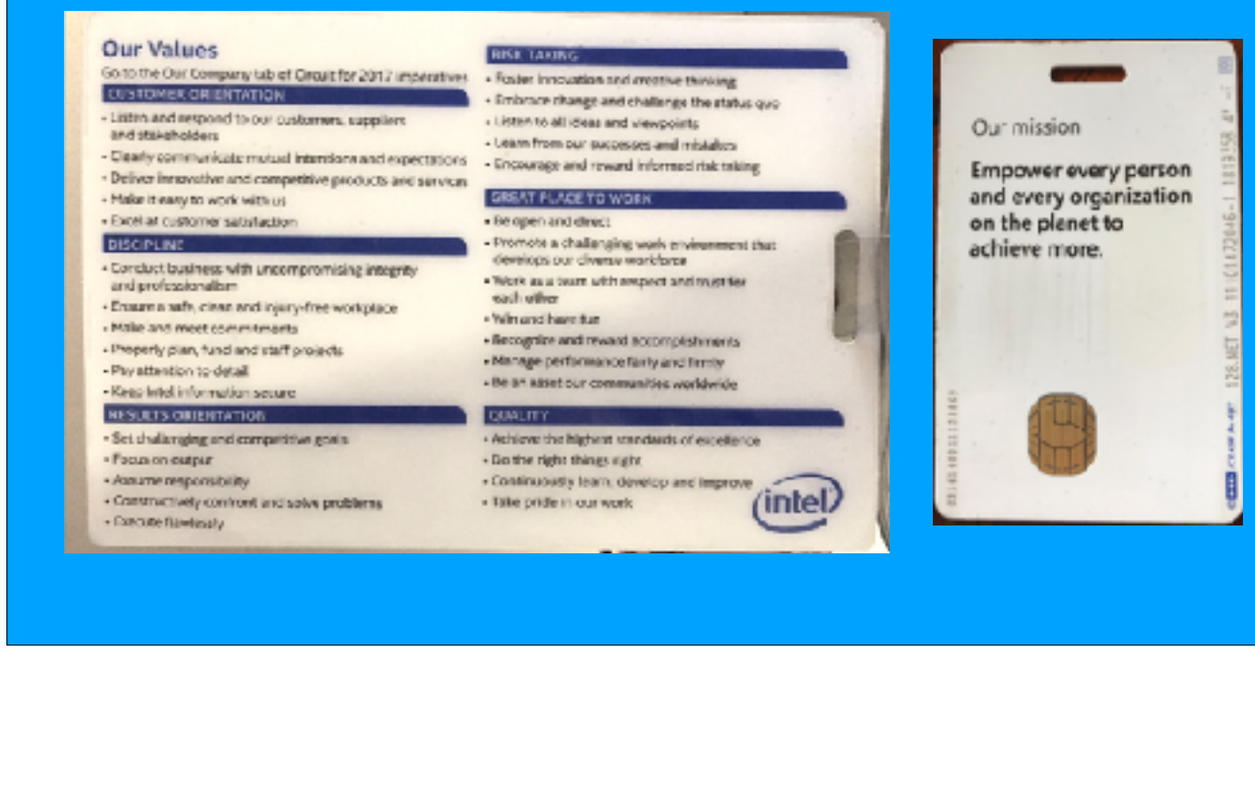
I’ve used ObjC here, but our “programming” language is unique to everyone, but they seem to have several things in common:
it is very bare-bones, no garbage collection, no ARC, no mechanisms to help with multi-threading
No KIS
It’s used in legacy code, in a very complex project setup
What’s SOLID?!
Single responsibility principle – a class should have only a single responsibility (i.e. only one potential change in the software's specification should be able to affect the specification of the class)
Open/closed principle – “software entities … should be open for extension, but closed for modification.” ..
Liskov substitution principle – “objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.” See also design by contract.
Interface segregation principle – “many client-specific interfaces are better than one general-purpose interface.”[8]
Dependency inversion principle – one should “depend upon abstractions, [not] concretions.”

The platform is heavily fragmented (we’re all different!) – we cannot just bump ourselves to the same version!
Too many nice tools!
All the keys, and all the values, but nothing is documented. Header files don’t contain any documentation, yet, everyone has been writing about it for years. It’s like stack overflow. Everyone is trying to figure things out, and once they think they’ve found a solution, they are posting about it. And then it get’s outdated. Or it doesn’t actually

Fight “Not invented here”?

So, I did what everyone would do, I had a look online. I mean, we have stack overflow, and such. There’s got to be a domain expert out there, right?!
“Core Values for Software Developers”

* It’s about code
* It’s about a company’s core values, or mission statement.
* It’s famous and/or rich people who have a pretty terrific life writing code, or architecting things, …
* Professional Blogs, books (there is actually one about “Softskills for Developers”)
* Personally, I’m not a spiritual person, but maybe it’s possible to ask someone with expertise in that field, your rabbi, vicar, friendly neighborhood witch and ask the vicar “hey, I wanna know about core values for software developers”.
* Why do I learn about personal hygiene as a kid, about clean code in college, but why does no one tell me about Core Value Hygiene?



So, I did what everyone would do, I had a look online. I mean, we have stack overflow, and such. There's got to be a domain expert out there, right?!
"Core Values for Software Developers"

* It's about code
* It's about a company's core values, or mission statement.
* It's famous and/or rich people who have a pretty terrific life writing code, or architecting things, …
* Professional Blogs, books (there is actually one about "Softskills for Developers")
* Personally, I'm not a spiritual person, but maybe it's possible to ask someone with expertise in that field, your rabbi, vicar, friendly neighborhood witch and ask the vicar "hey, I wanna know about core values for software developers".
* Why do I learn about personal hygiene as a kid, about clean code in college, but why does no one tell me about Core Value Hygiene?

When dealing with complicated, stateful systems, dutiful book-keeping is essential for maintaining sanity.

Eventually, I ended up reading that NSHipster-blogpost again.

Matt Thompson, in a post on NSHipster in 2013
http://nshipster.com/key-value-observing/

```
@property (nonatomic) NSDictionary<NSString *, Value *>* coreValues;
```

Which values?!

Value.h doesn’t actually tell us what keys there are! And what “Value” actually is.

Fuzzing

I needed to do some fuzzing, to find out what values life.framework

But...how?!

Talk to as many peers in my community as possible.
Read the stupid “how I work” things. Read the stories about failure. Read twitter. Talk to old people. Listen to podcasts (very rarely) -> admit not doing that one!.

I’ve been doing trial and error things for the past 8-9 years. I think I have figured out a few things over the years but I constantly fail.

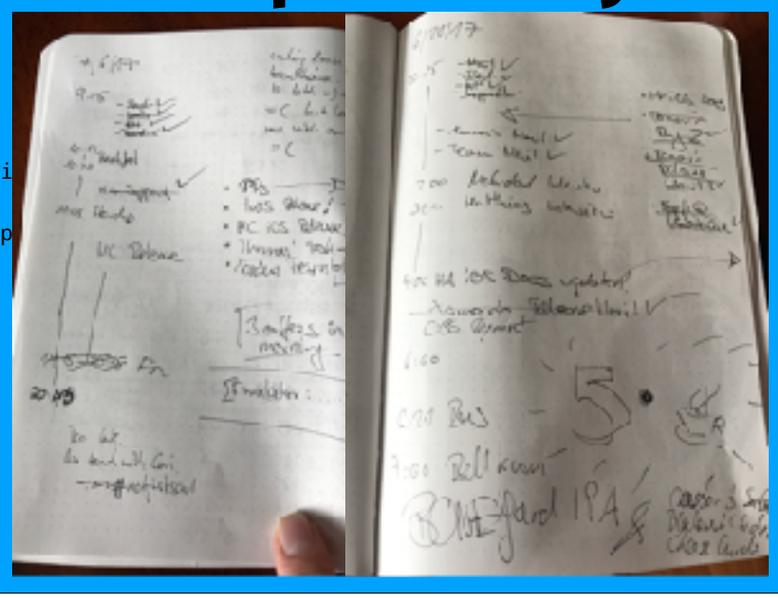
Things I do:
Practicing Active listening
mentoring, reading
Meditation, Yoga, Working out, Regimen, Reading, Communication
…

Responsibility

```
- (voi

// Imp

}
```



Sustainable hours!
Sustainable lifestyle!

This is where I do actual bookkeeping, my DAILY PLANNER.
I wasn’t very happy with it for a while, because I wanted it to look nice, to be “pretty”.
It doesn’t have to be pretty. It has a job.

a) remind me of what I accomplished every day
b) it requires attention and helps me being Aware!
c) It makes me be cautious about the time I put into my work
d) It reminds me to eat!

I learned to use that when I was in this dark places. Writing down things. I need that haptic feedback. I didn’t actually do that continuously. I sometimes take a break from writing things down. I’m on my longest streak, since fall last year. It’s okay, to skip a day.

Diversity / Inclusion

```
- (BOOL)perceiveMyselfAsInclusive {
 return YES;
}

- (NSArray <Actions> *) doInclusiveShit {

// Implementation
 return nil;
}
```

This is one that I didn’t "get right” for a long time, I’m still horribly wrong sometimes. I’m learning. It’s actually easy of you to say your own experience is different than the other, so the other person is wrong.
Women’s wages
Why do transgender friends of mine not tell people at work that they are transgender?

My team is 22% female, it was almost 40% before we merged with a male-only team. We do have 0 African American folks in the team but a lot of other people of color. Why?!

Separate the truth from the BS! Ask open questions. Be critical.

Inclusion is easier as it seems: “Hidden Figures” -> scene with the lead astronaut!

Can I ask my manager if they try to hire female devs, or people of color? Absolutely! Did I do it? Yes!
Can I try and find someone and refer them? Absolutely! Did I do it? No.
Can I bring this up in an employee survey? Sure! Did I do it? Yes?
What I did is: start using THEY all the time. People actually start using it, too. And speak out and ask people to use it. It happened to me the other day in the cocoapods slack -> I got pinged and asked to use gender neutral language.

If diversity is not important to you, own that!
Educate yourself.

Empathy

```
- (NSNumber)currentEmpathyLevel {
 return @1; //should be 10?
}


- (void)improveEmpathy {
 // how?
}
```

Especially when working remotely!
When in a new team, you don’t know each other
when getting reorg’d
Important for PRs -> don’t be a dick! Sometimes language is the problem (don’t assume English is easy for the person on the other side). Maybe they have a really hard time right now, and you just don’t know it.
Practice empathy:
active listening
reading

Daring

```
- (void)dareToBe:(Feeling *) aFeeling {
}

- (void)dareToDo:(Thing *) aScaryThing {
}
```

Dare - to
Curiosity
Courage, be open for change, be open to open up to someone

Dare to think outside your lunchbox! There’s more to lunch than sandwiches!
I'm not very good at it, but: I’m daring myself to something, and then treat myself to something.
I date to run further, and reward myself with something. This experience helps me with work-related “challenges”.
This whole talk was a daring thing….I mentioned my impostor syndrome in the beginning.

Failure

```
- (void)do {
  @try {
    // Code that can potentially throw an exception
  } @catch (NSException *exception) {
    // Handle an exception thrown in the @try block
  } @finally {
    // Code that gets executed whether or not an exception is thrown
  }
}
```

How can we have try/catch in our code but not in our lives?!
Is failure ok?
Do we hide failure or do we admit it?
Do we cherish failure?
Do we really allow failure? Is it more than words?

What happened when the idea of this talk didn’t make it into UIKonf?

Coping with failure: I do yoga. I talk to my wife. I think about what I can do find out if it’s still worth pursuing.

Authenticity

```
- (void)beAwesomeCoworker {
 [super beAwesomeCoworker];

 self.personality = [WorkPersonality new];

 // vs.

 self.personality = [self beYourself];

 // more…
}
```

Story about refused job…

Community

This is our culture, we own it.
We should act a little more like it and offer it some protection.

Well, we all here value community, because we all show up at Xcoders. So that’s one of the things we value a lot. So,

If it’s within a team at work, with our work family, if it’s when meeting with customers, if it’s when meeting on a Thursday for Xcoders, they are all their own communities.
And they are all part of a certain culture, our culture.
This is our culture, own it, we should act a little more like it and offer it some protection.
(By my friend and tattoo artist)

-> e.g. what happened in the Cocoapods slack, or what the JS community does for events (they were one of the first communities to establish a code of conduct)