

Confessions of an Agile Addict

Ole Friis Østergaard

Silverbullet[®]

of@silverbullet.dk
[@olefriis](https://twitter.com/olefriis)
www.silverbullet.dk

Me?

iOS

Grails

JavaScript

Java

**Programmer interested in
good development
practices**

Android

Sinatra

Groovy

Ruby

Angular

Rails

SCRUM

Kanban

HTML5

TDD

Not me?

Thought leader

Drug addict

Extrovert

Famous

Expert on everything

Seasoned speaker

Psychiatrist

Native English-speaking

You

Agile

Want to enjoy your job

Developer/ Programmer

Want to evolve

Want to learn

The point?

Get addicted to the right stuff.

Love your work.

Feel great about yourself.

Be agile.

The plan

What's an addiction?

Agile practices

Agile addictions

Me! Me! Me!

Session Code

3374

Addictions



Silverbullet[•]

About addictions

There are habits and addictions.

Habit: Something you "just do".

Addiction: You feel it in your guts if you're missing it for too long.

About addictions

There are habits and addictions.

Habit: Something you "just do".

Addiction: You feel it in your guts if you're missing it for too long.

Impaired control, continued use despite consequences, and denial...

Examples

Drugs

Examples

Running

Examples

Sex



Image from Wikipedia

Agile Practices



Silverbullet[•]

Test-Driven Development

Specify before you code

Makes you write fewer bugs

Helps you design better systems

Gives you tools to document behaviour

Helps managing risk when refactoring

Still a very rare sight (at least in Denmark)

Doing Retrospectives

You and your team sit down an hour or two and discuss what happened in the past sprint

You focus on getting a little bit better in the next sprint, by not repeating past mistakes and by strengthening your successes

Maybe you add in an experiment to the coming sprint?

Pair Programming

You pair up with a teammate

One at the keyboard, another "driving"

Leads to better design

Leads to more shared knowledge

Leads to better personal interactions on team

Leads to "less cheating"

Cross-Functional Teams

There's no "one guy" who knows how to do one thing

Everybody can join in on different parts of the system

Speaking to customers

"Agile" is mostly about feedback

Ultimately, the best feedback is from whomever is going to use your product

The sooner, the better

Not everybody agrees

Tests are in principle "muda" (waste)

Badly written tests are of no value

Badly written tests are a pain to refactor

The "beyond TDD" movement

Pair programming takes two instead of one

Pair programming requires communication skills

Pair programming "is not for everybody"

Customer doesn't always see the whole picture

And this is not all

Focusing on flow
Systems thinking
Continuous Integration

...

About Cross-Functional Teams...

You **do** know what this is, and how to fix it, right?

```
<<<<<<< HEAD:my_program.rb  
  
if (is_ok)  
=====  
if (!is_invalid)  
>>>>>>> my_branch:my_program.rb
```

How Does That Relate?!?

Be Agile for your Own Sake

Silverbullet[®]

You hate code reviews...

It's like meetings: They split up your day
They rip you out of context
You don't really take the time necessary to
understand all the code and tests

And, actually, it's often **too late** to fix the
underlying problems with the code

...so you pair program!

Code review is part of the deal

You hate bad atmosphere...

Colleagues who don't speak together

Not feeling welcome in the office

...so you do retrospectives!

An amazing way to get things going smoothly

More often than not, there are simple, underlying reasons for problems on a team

Just talking about it helps - fixing it is even better

From a famous guy...

"My wife has often said that I'm creatively co-dependant and that basically, in order for my creative mind to be engaged at 100%, I need everybody in the vicinity to be in a good mood, to be happy and to be comfortable; negative energy pulls on me creatively.

So if someone's uncomfortable or having a bad day in the room, I feel the need to perk that person up and try to make them laugh. It seems like this is a beautiful, Jesus-like, selfless act, when in actuality, a huge part of it is very selfish and needy on my part"

- Will Smith

<http://www.darkhorizons.com/features/1312/will-smith-for-men-in-black-2>

Image from Wikipedia



You're addicted to positive feedback...

Isn't that what we go to work for?

We're simple creatures

...so we speak to customers!

Surely, not everything from them sounds sweet

But try starting a dialog, and you'll find that they are extremely thankful for the simple fact that you listen to them

Often very simple improvements make your customer the sweetest person in the world

You're a social being...

Sitting alone in a cubicle is a sad life

You get no input, you don't evolve

...so you pair up!

Pair programming is for all social beings

You get to know your colleagues much better

You fight, you cry, you laugh... and get loads of input all the time

You're addicted to hot-fixing

Gives fame and kudos saving the production systems from disaster

The boss considers you a vital resource

You're a hero!

It's a great rush

...so you write crappy code

Hot fixes means doing a quick'n'dirty job

Being the only person alive who can mend the code gives a bit of status

...and if it works, why change?

You're afraid of critique...

Critique hurts

Customers don't always understand the deeper workings of software

Your colleagues have other ways of doing stuff

...so you don't speak to customers

Their opinion often gets in the way

It takes a lot of effort to explain your point of view

...and you don't pair program

There's so much arguing going on

You'll lose half of your discussions

Me! Me! Me!

Silverbullet[•]

Haven't been agile all my life...

Start of my career: Waterfall-only company

Felt bad about some things, but thought this was the way to be professional

Got introduced later to agile practices, and it resonated with me. Why?

Becoming aware of my own addictions

We all have a number of addictions

Finding out which takes reflection

I've thought a bit about what a sad person I am

I die if I sit by myself in a corner!

Would never be able to work from home (regularly)

Need to chat

Need to discuss ideas and solutions

...so I pair program

Gives me so much professional input

I learn something new each day

Keeps my motivation going

...and turns colleagues into friends

When starting a new task...

...I always start by asking "How do I test this?"

After several years practising TDD, I've become addicted

It just feels plain wrong to edit code without tests - my guts twist

TDD and flossing your teeth

Testing is like flossing your teeth (Kent Beck) - you might not get the full benefit right away, but in time you'll be sorry if you don't

I floss my teeth, and it's become a habit/addiction just like brushing my teeth

TDD vs. slot machines

"Traditional coding" has been compared to sitting by a slot machine

You seldom win, but when you do, your brain releases enough endorphins that you'll keep going

TDD at first seems slower, but keeps the rhythm going

Messy kitchen

I cannot work in a messy kitchen

Need to clean up before cooking

Need to clean up while cooking

Feel claustrophobic otherwise

Same with code

Controlling your Addictions

Silverbullet[•]

Embrace change...

You want to get rid of your bad addictions

Keep your good addictions

And maybe collect a few good addictions?

First step: Reflect & Admit

Do any of the mentioned addictions resonate with you?

Should they?

Is there anything you'd like to change?

Then... practice!!

It takes time to change your perception

It takes persistence

Some like "cold turkeys", some like gradual change

Be open about it, tell your colleagues!!

Luckily...

There is no reason to be embarrassed about your "agile addictions".

Some of my addictions

Running

Fruit

Test-Driven Development

Coffee

Flossing my teeth

Continuous Integration

Pair Programming

Being Cross-Functional

Winter Swimming

Vitamin D

**Agile: I do it for
myself!**

Silverbullet[•]

But remember...

Don't Do Drugs!

Silverbullet[•]

of@silverbullet.dk
@olefriis
www.silverbullet.dk

A Few References

Kent Beck: Ease at Work: <http://www.youtube.com/playlist?list=PLE9763518A2765373>

100% time: <http://cemerick.com/2013/02/15/100-time/>

Programmers and Chefs: <http://www.codinghorror.com/blog/2006/05/programmers-and-chefs.html>