

Accelerating Agile

- an experience report

Dan North

@tastapod

Dan North & Associates

Once upon a time...

@tastapod

Step 1: Learn the domain

- ✓ **Seed** the team with a domain expert
- ✓ **Study** trading – just like a trader!
- ✓ **Practise** trading with the traders
and while you're there...

Step 2: Prioritise **risky** over **valuable**

- ✓ Actively surface uncertainty
- ✓ Domain uncertainty
 - Integrating with a trading exchange
 - Managing orders
- ✓ Technical uncertainty
 - Latency, Throughput
 - Toolchain

Step 3: Plan as far as you need

- ✓ **Adjust** as you learn
- ✓ **Reset** the board
 - Every week? Every day?
- ✓ **Review** your planning horizon

Step 4: Try something different

Languages

- Scala
- JavaScript and nodejs
- Erlang, Clojure, Go

Programming styles

- CSP
- Actors
- “Turn-based processing”
- Fork-join

Step 5: Fire, Aim, Ready

- ✓ Get something (anything!) in front of **users**
- ✓ The best **feedback** is from real use
- ✓ **Showcase** frequently – even daily!

Step 6: Build **small**, separate pieces

“Share memory by communicating”

DRY is the enemy of **decoupled**

Don't be afraid of functions
...languages or libraries

Step 7: Deploy **small**, separate pieces

- ✓ Make component deployment **quick**
- ✓ Make product deployment **consistent**
- ✓ Make components **self-describing**
- ✓ Make environments **unsurprising**

Step 8: Prefer **simple** over **easy**

I'm using Java. I'm writing HTTP-based services. Do I really need a servlet container?

- <https://github.com/webbit/webbit>

I need to manage binary dependencies. Do I really need an XML-based Internet downloader?

- <https://github.com/mfoemmel/git>

How hard does monitoring really need to be?

- Idea to wireframe to working implementation *in a morning!*

Step 9: Make the trade-offs

Build vs. buy vs. OSS

Learning a framework vs. rolling your own

Does logging really need a “framework”?

Step 10: Share the love

- ✓ Pairing
- ✓ Learning lunches
- ✓ Code review (!!)
- ✓ On-boarding

Step 11: Be ok with “failure”

Product Development not Project Delivery

Progress is a series of experiments

Failing fast is succeeding!

Step 12: There are always 12 steps

Delivering this fast can be **addictive** :)

It can also cause feelings of **euphoria**

There are probably groups you can talk to

Thanks for listening

dan@dannorth.net

<http://dannorth.net>

And big thanks to my former team at DRW

@tastapod