

# Sharpening the tools

Dan North, DRW



# How you become a software craftsman

## The Dreyfus Model

Novice programmer

Advanced beginner

Competent

Proficient

Expert

...

Novice architect...

# Along the way you gather some tools

## Technical tools

- Languages and libraries
- IDEs, Operating Systems
- SCM and build tools
- “Back pocket” utilities

## Programming tools

- Refactoring
- TDD
- Patterns

## Modelling tools

- UML, DDD
- Lo-fi prototyping

## Methodology tools

- XP, Scrum, BDD
- Lean, Kanban, ToC

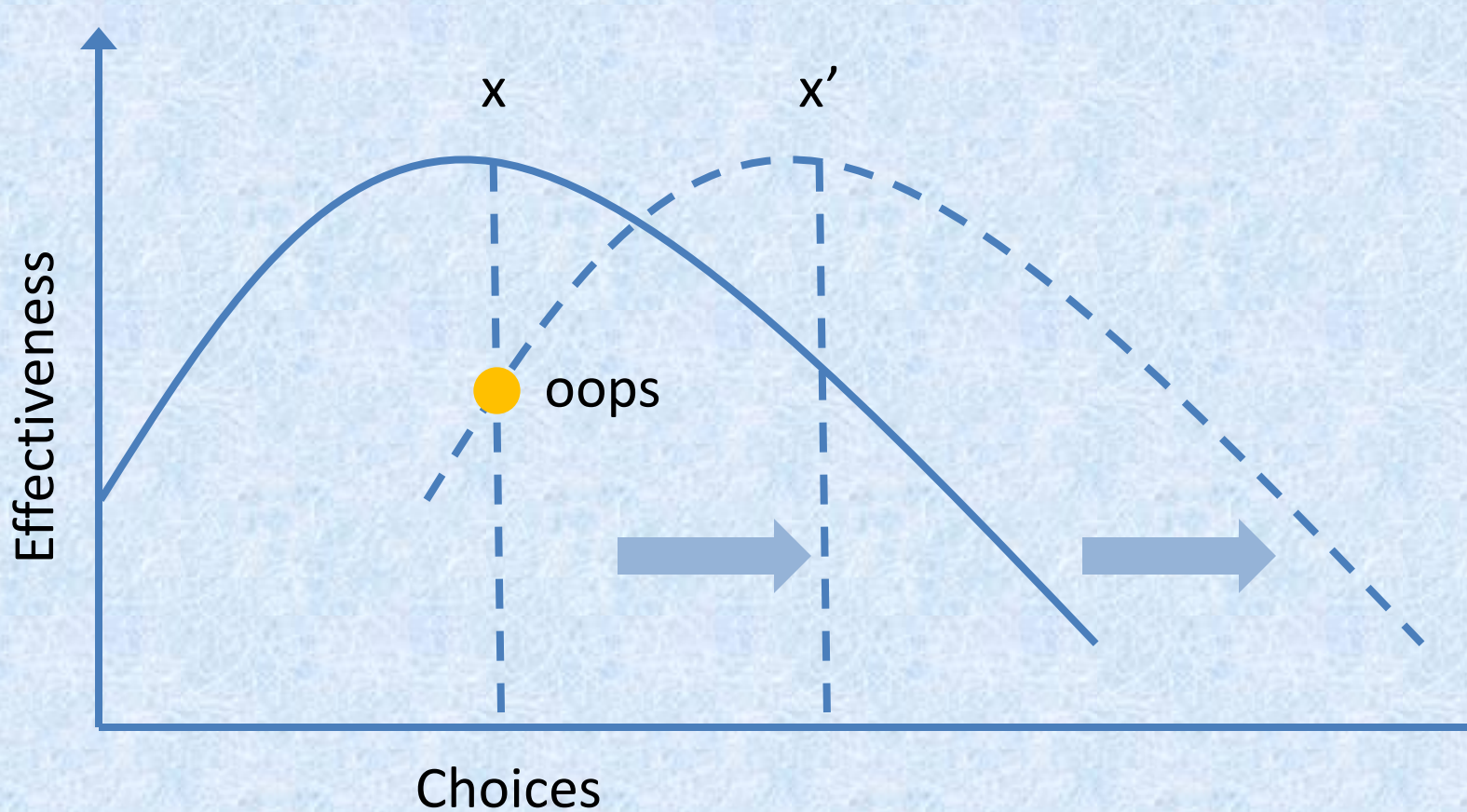
## Personal tools

- To-do lists, GTD, Pomodoro
- GROW framework
- NLP “chunking”

## Group tools

- Six thinking hats
- Stand-ups
- Retrospectives

# Meanwhile...



*John Roberts – the Modern Firm*

# Then one day I got a shock

I hadn't been "doing" software for a while...

or any *real* coding

but surely it's like riding a bike?

*It's scary how quickly your skills can atrophy*

# Tip #1: Practise the basics

Practise kata

Code for fun!

Learn a new language *that you don't need!*

Learn shortcuts

Practise your shell-fu

# Tip #2: Learn from other people

Stalk experts

Listen to novices

Read books, blogs and online magazines

*Read code*

Follow people on Twitter

Programme in pairs

# Tip #3: Understand trends

Build a network of people you listen to

This time last year I knew nothing about

- distributed SCM
- Clojure, Erlang and Scala
- HTML 5, WebSockets, JavaScript, V8
- NOSQL data stores

This year I know *nearly* nothing

*Do you know what you don't know?*

# Tip #4: Share knowledge

Blog about your successes – and your failures  
– You never know what people might find useful

Join mailing lists

– Don't just lurk, answer questions

Organise learning lunches

– or Muffin Mornings

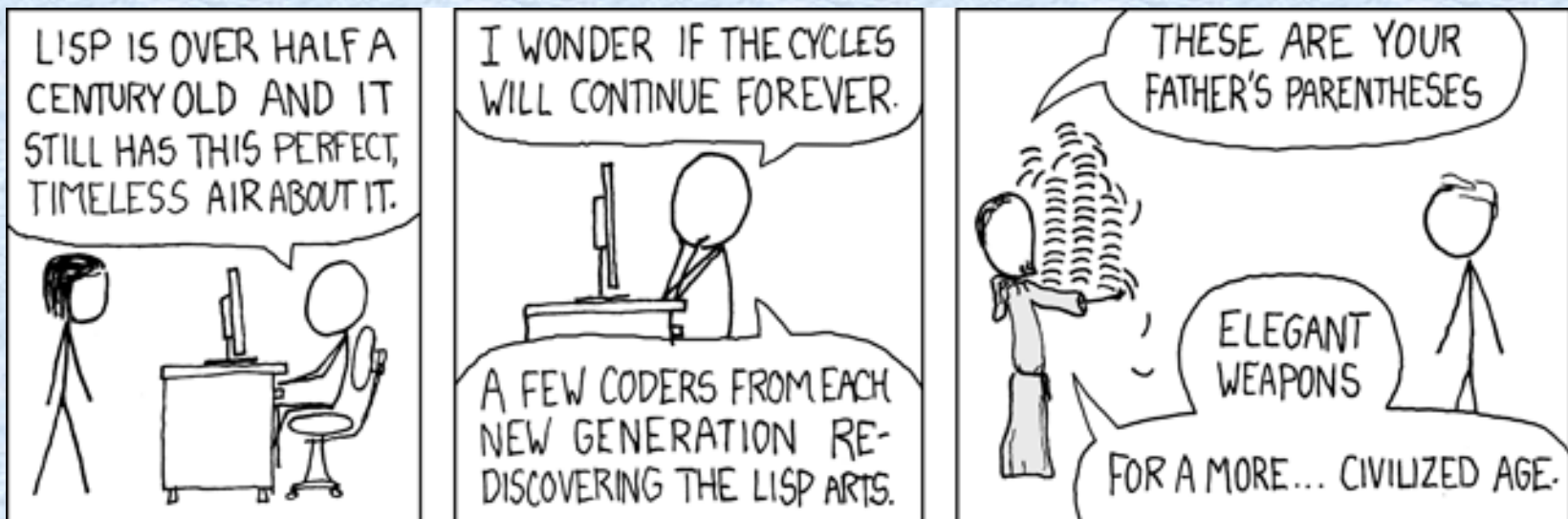
Speak at conferences

# Tip #5: Maintain your toolbox

Some tools are timeless, some are disposable

Some tools are a delight to rediscover

Use the “clothing” rule to update your tools



<http://xkcd.com/297/>

# Tip #6: Learn how to learn

Understand how learning works

Understand second-order incompetence

Start using Six Thinking Hats

Practise chunking

# And then do it all again!

Eat your own dogfood

Actively solicit feedback

“Listen like you don’t know the answer”

# Conclusion

Always assume you are out of date

- Never stop gathering and sharpening your tools

You owe it to yourself to keep current

- as well as to your team and your organisation

*It's a continual cycle of learning and relearning*

# Thank you

[dnorth@drwuk.com](mailto:dnorth@drwuk.com)

[@tastapod](#)

<http://dannorth.net>