



xpert7

Mythical
Developer
Productivity

Erik Dörnenburg
Software Developer
ThoughtWorks, Inc.

Presented by Cornerstone

About me

- a technologist at ThoughtWorks in the UK
- writing code *and* doing architecture work
- ThoughtWorks delivers complex projects using Agile methods
- has about 750 people in Australia, Canada, China, India, and the UK and US
- More at erik.doernenburg.com



ThoughtWorks[®]
The art of heavy lifting.[™]

Cornerstone Slide 2

What *are* we doing?

- Software Development is a very young field, in comparison to
 - constructing buildings and bridges
 - or ships and ports
 - or making cabinets or vases
- We're trying to use metaphors
 - *architects and engineers*
 - *the build process*
 - *The Art of Computer Programming*

Cornerstone Slide 3

Do we need a new approach?

- *From idea to realisation* at an unprecedented speed
- We have executable models
 - so we can all be architects!
- We have *foreach*
 - Have you seen a real architecture plan?
- We don't really know what we are doing
 - Can you imagine 1% contingency?
- So, when we look at what increases productivity in other disciplines, how much can we copy?

Cornerstone Slide 4

Myth #10

“Every developer needs their own office!”

**So that they can work undisturbed...
...and communicate less.**

- If we were working in such a way that we can be on our own most of the time, why shouldn't our work be sent off-shore?
- Most problems are best solved by a team working closely together.

Cornerstone Slide 5

Myth #9

“We must re-use our objects!”

**So that we save time later...
...but we just add the abstraction now.**

- Re-using objects can save time
- but adding abstractions *just in case* is wasteful
- it usually increases coupling between components as well

Cornerstone Slide 6

Myth #8

2007: “We need a Software Factory!”

So that we can develop faster...
...once we’ve finished the framework.

- Achieving *economics of scale* by abstractions can be really costly
- Providing the right extension points requires experience in the specific field

Cornerstone Slide 7

Myth #7

“We need transparent distribution!”

So that we don’t waste our time writing
marshalling code...

“We’ll have at least 5 message types!”
– Anonymous developer

- Method calls are: fast and reliable
- Networks are: slow and unreliable
- Isn’t RPC/RMI/Remoting somewhat confusing?

“95% transparent is not good enough. In fact,
it is worse because it deceives developers.”
– Werner Vogels

Cornerstone Slide 8

Myth #6

“We must keep our options open!”

So that everything is configurable

- Programming in XML is programming
- Redeploying a new version should be trivial

So that swapping implementations is easy

- Not when we built on abstractions already

So that I can split the stack to scale out

- This is a self-fulfilling prophecy...

Cornerstone Slide 9

Myth #5

“All I need is a book!”

So that I can learn new technologies and processes.

- Many authors describe what worked. For them.
- Many books describe what something is but not how it is applied.

Beware of cargo cults!

Cornerstone Slide 10

Myth #4

“Pair programming slows down development!”

“If typing was the limiting factor...”
— Martin Fowler

- Software development is a creative activity
 - Most people need to bounce-off ideas
- Goals beyond just writing *an* implementation:
 - consistency
 - re-use (yes!)
 - sustainability (or: the bus problem)

Cornerstone Slide 11

Myth #3

“We have to program in diagrams!”

“I’ll take an if statement over an action diagram any day.”
— Martin Fowler

- A picture says more than a thousand words
- but it can be cumbersome to create
- and we can generate diagrams from text

- Abstractions can provide order-of-magnitude changes, but different representations cannot

Cornerstone Slide 12

Myth #2

“I must have graphical tools!”

Because I *like* searching for the database name in an explorer view

- Graphical tools are good for exploration but most of the time we know what we want

Because I *like* to repeat the same 10 clicks each time I deploy my EAR file

- Scripts are more suitable for repetitive tasks

Cornerstone

Slide 13

Myth #1

“I know my IDE!”

Ok, not every menu item...

All the keyboard shortcuts?!

- Programming is not typing in code
- it is working with code
- realising ideas, e.g. creating a Map object
- applying refactorings
- navigating the code-base

Cornerstone

Slide 14

I have one more

“Using X will make us more productive!”

- Good teams will look around and find what works best for them
- and they often write about it
- Beware of cargo cults!
- Correlation is not the same as causality

“XP is a process that can make good teams much much more successful. Same goes with Ruby on Rails.”
– Jon Tirsén

Cornerstone Slide 15

The End

Thank you!

erik@thoughtworks.com

Cornerstone Slide 16