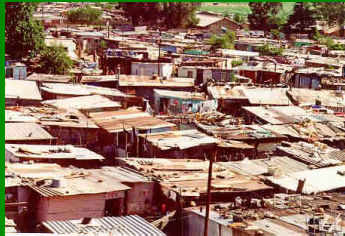


Big Balls of Mud in Agile Development – Can we Avoid Them?

Joseph W. Yoder



Copyright 2010 Joseph W. Yoder & The Refactory, Inc.



Evolved from The UIUC SAG

In the early 90's we were studying objects, frameworks, components, reusability, patterns, "good" architecture.

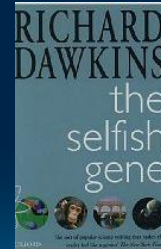
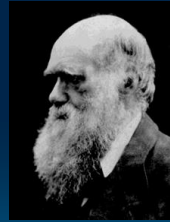


However, in our SAG group we often noticed that although we talk a good game, many successful systems do not have a good internal structure at all.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 2

Selfish Class

Brian and I had just published a paper called *Selfish Class* which takes a *code's-eye view of software reuse and evolution*.



In contrast, our BBoM paper noted that in reality, a lot of code was hard to (re)-use.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 3

Why BBoM?

Why was this phenomenon so prevalent in our industry? We talk a good game.

We had seen where Lisp had failed, Smalltalk was starting to fail, Windows was winning. Why was this? What is there about some systems that failed compared to systems that succeed, even when they seemed better.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 4

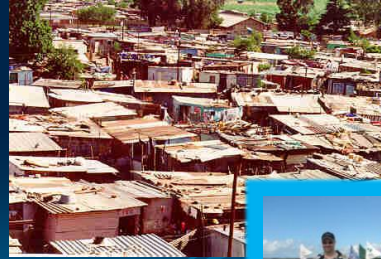
Big Ball of Mud

Alias: Shantytown, Spaghetti Code



A BIG BALL OF MUD is haphazardly structured, sprawling, sloppy, duct-tape and bailing wire, spaghetti code jungle.

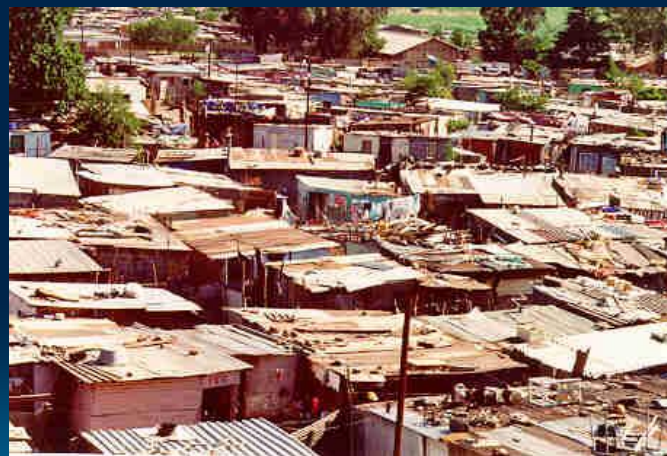
The de-facto standard software architecture. Why is the gap between what we **preach** and what we **practice** so large?



We preach we want to build high quality systems but why are BBoMs so prevalent?

Big Balls of Mud in Agile Development – Can we Avoid Them -- 5

Big Ball of Mud



Big Balls of Mud in Agile Development – Can we Avoid Them -- 6

Worse is Better

Idea resembles Gabriel's 1991
"Worse is Better"

Worse is Better is an argument to release early and then have the market help you design the final product. It is taken as the first published argument for open source, among other things.

Do BBoM systems have a Quality?

Big Balls of Mud in Agile Development – Can we Avoid Them -- 7

Worse is Better (examples)

Betamax vs VHS Format

- Why did VHS win?
- Betamax was arguably a better format

Macintosh vs Windows

- Mac was easier to use
- Far superior in many ways

MS Word/Publisher vs FrameMaker

- Lot's of people use word
- FrameMaker is better for books

Big Balls of Mud in Agile Development – Can we Avoid Them -- 8

What exactly do we mean by "Big"?

Well, for teams I consider $> 10^2$ big
and for code I consider $> 10^5$ big

Teams can write good code. Smalltalk is an example. I've seen teams of things written by 10^1 or 10^2 be pretty good and definitely would not be considered to be a BBoM.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 9

Legacy == Mud?



Big Balls of Mud in Agile Development – Can we Avoid Them -- 10

Legacy != Mud???

Does Legacy happen within months or a year after the first release?

Or is legacy after the second release?

What about Muddy code that is released on the first version? Is this a counterexample?

Is all Legacy Mud? Smalltalk???

Big Balls of Mud in Agile Development – Can we Avoid Them -- 11

Is Mud Normal?

Well, just read our paper....there are "normal" reasons why it happens. Maybe it is the best we can do right now.

If mud is such a bad thing, why do people keep making it?

Maybe if we accept it and teach it more then we can deal with it better and help prevent it from getting too bad.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 12

Question

So, why DO people build Big Balls of Mud?

Where Mud Comes From



DILBERT © United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited.

People Write Code → People make Mud

Big Balls of Mud in Agile Development – Can we Avoid Them -- 14

Keep it Working, Piecemeal Growth, Throwaway Code



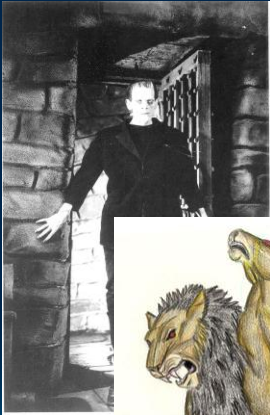
Big Balls of Mud in Agile Development – Can we Avoid Them -- 15

Copy 'n' Paste



Big Balls of Mud in Agile Development – Can we Avoid Them -- 16

The Age of Sampling



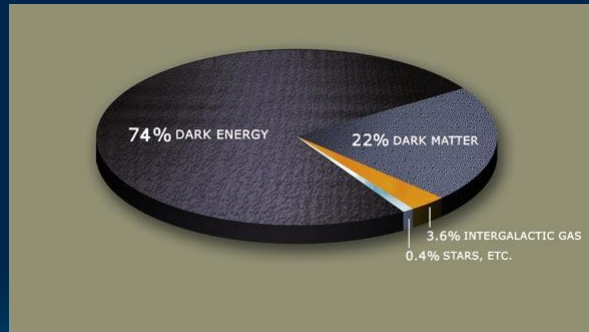
Big Balls of Mud in Agile Development – Can we Avoid Them -- 17

Big Bucket of Glue



Big Balls of Mud in Agile Development – Can we Avoid Them -- 18

The Mystery of Dark Matter



Accidental Complexity??? Maybe our current state of the art leads to Mud!

Big Balls of Mud in Agile Development – Can we Avoid Them -- 19

They Have a Name



Big Balls of Mud in Agile Development – Can we Avoid Them -- 20

Agile to the Rescue???

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

...From the Agile Manifesto

Big Balls of Mud in Agile Development – Can we Avoid Them -- 21

Question

What Agile Practices help us avoid or cope with mud? Does Agile practices such as TDD really help minimize mud? What are we doing RIGHT?

What Agile Practices contribute to the problem? Encourage mud? So Is Mud really the best that Agile can do?

Can Agile Help?

Scrum, TDD, Refactoring, Regular
Feedback, Testing, More Eyes, ...

Good People!

Continuous attention to technical excellence!

Retrospectives!

Face-To-Face conversation.

Motivated individuals with the environment
and support they need.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 23

Do Some Agile Principles Encourage mud?

Lack of Upfront Design?

Late changes to the requirements
of the system?

Continuously Evolving the Architecture?

Piecemeal Growth?

Focus on Process rather than Architecture?

Working code is the measure of success!

I'm sure there are more!!!

Big Balls of Mud in Agile Development – Can we Avoid Them -- 24

Question

Is Craftsmanship the Cure? Or maybe it is the problem? Is ascribing poor code to unhygienic habits, even malpractice enough, or naive; or is mud inevitable?

Does quality matter? Does quality just get in the way? Is "Clean" the Answer? Or only part of the answer? Or a sideshow?

Quality

Quality Definition: a peculiar and essential character or nature, an inherent feature or property, a degree of excellence or grade, a distinguishing attribute or characteristic



Big Balls of Mud in Agile Development – Can we Avoid Them -- 26

Quality (Whose perspective)

Artist	Scientist
important/boring	true/false
Designer	Engineer
cool/uncool	good/bad

"The Four Winds of Making"...Gabriel

Does quality on the inside mean quality on the outside?

Rich Gold "The Plenitude: Creativity, Innovation & Making Stuff (Simplicity: Design, Technology, Business, Life)"
Triggers and Practices – Richard Gabriel <http://www.dreamsongs.com>

Big Balls of Mud in Agile Development – Can we Avoid Them -- 27

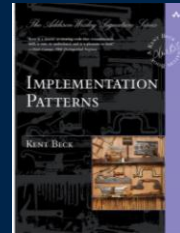
Being Good Enough

- Quality of being good enough
- Does it meet the minimum requirements
- Quality has many competing forces...are we designing a system for online orders or for controlling the space shuttle, they have different qualities, thus different patterns and solutions apply
- Perfection is the enemy of **Good Enough**
- Maybe Quality without a Number

Big Balls of Mud in Agile Development – Can we Avoid Them -- 28

Implementation Patterns

Patterns about creating quality code that communicates well, is easy to understand, and is a pleasure to read. Book is about patterns of “Quality” code.



But...Kent states, “...*this book is built on a fragile premise: that good code matters. I’ve seen too much ugly code make too much money to believe that quality of code is either necessary or sufficient for commercial success or widespread use. However I still believe quality of code matters.*”

Patterns assist with making code more bug free and easier to maintain and extend.

Big Balls of Mud in Agile Development – Can we Avoid Them -- 29

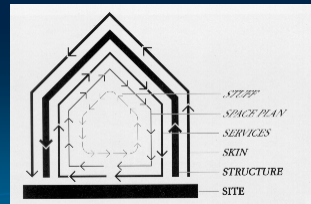
Some Answers to Mud!?!

We can gentrify, rehabilitate, or make-over code helping clean up the mud.

Even some patterns, frameworks, components, and objects helped with mud. Agile has helped some.

Stuart Brand's Shearing Layers

- Buildings are a set of components that evolve in different timescales.
- Layers: site, structure, skin, services, space plan, stuff. Each layer has its own value, and speed of change (pace).
- Buildings adapt because faster layers (services) are not obstructed by slower ones (structure).

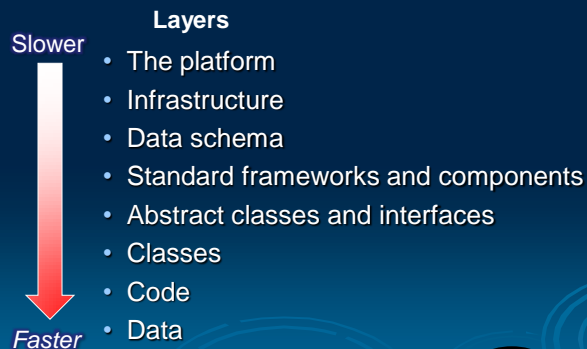


—Stuart Brand, *How Buildings Learn*

Big Balls of Mud in Agile Development – Can we Avoid Them -- 31

Yoder and Foote's Software Shearing Layers

“Factor your system so that artifacts that change at similar rates are together.”—Foote & Yoder, *Ball of Mud*, PLoPD4



Big Balls of Mud in Agile Development – Can we Avoid Them -- 32

Total Code Makeover



Can we just Refactor out of Mud?

Sweep the Mess Under the Rug?

Big Balls of Mud in Agile Development – Can we Avoid Them -- 33

Sweep It Under the Rug



Cover it up to keep other areas clean
(Façade and other Wrapper Patterns)

Big Balls of Mud in Agile Development – Can we Avoid Them -- 34

Put a rug at the Front Door

Protect Important Components!

Keep other parts of the system clean.

Sometimes Glue code (Mediators) helps
keep others parts of the system cleaner.
(Anti-Corruption Layer -- Eric Evans)

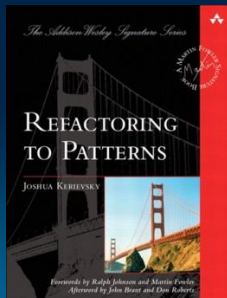


Big Balls of Mud in Agile Development – Can we Avoid Them -- 35

Code Make Over

Refactoring can help reverse some mud. The
tradeoff is cost and time....maybe with technology.

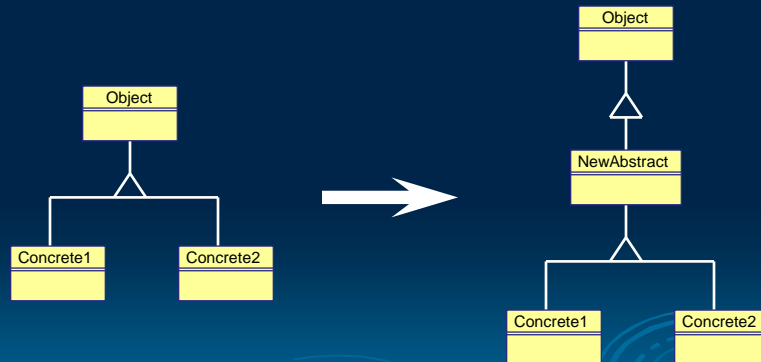
Refactoring to Better Design (Patterns)



Big Balls of Mud in Agile Development – Can we Avoid Them -- 36

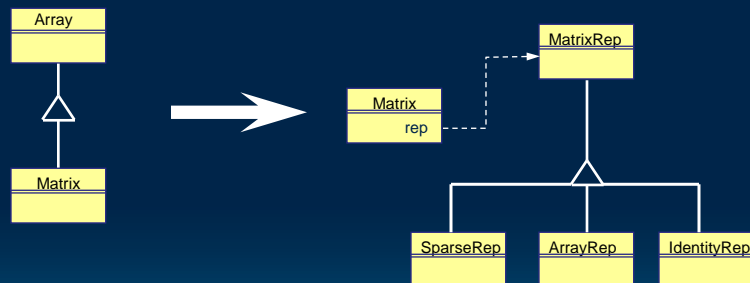
A Simple Refactoring

Create Empty Class



Adapted from Don Roberts, The Refactory, Inc.
Big Balls of Mud in Agile Development – Can we Avoid Them -- 37

A Complex Refactoring



Refactoring can be hard but there are a lot of
small steps that lead to big gains in mud busting

Adapted from Don Roberts, The Refactory, Inc.
Big Balls of Mud in Agile Development – Can we Avoid Them -- 38

Tools Can Help!

What is the role of tools in draining these swamps?

Refactoring Tools, Testing Tools, XUnit, Lint Tools, Code Critics, ...

Tools can help, but too often too much is put on tools as the solution (silver bullet).

Still requires lots of hard work and good people using the right tools!

Big Balls of Mud in Agile Development – Can we Avoid Them -- 39

Testing



Big Balls of Mud in Agile Development – Can we Avoid Them -- 40

Good Design

It is a myth that Agile doesn't support Design....It is ok to have good Design!

Good proven practices and patterns can help....It's ok to think!!!

Leave the code cleaner than when you came there (Leave the Campground Clean -- Craftmanship).

Big Balls of Mud in Agile Development – Can we Avoid Them -- 41

Many Quality Patterns Written

Design Patterns
Patterns for Fault Tolerant Software
Performance Patterns
Small Memory Software Patterns
Analysis Patterns
Security Patterns
Stability Patterns
Usability Patterns

.....

Imitate or use proven quality techniques

<http://www.hillside.net>

Big Balls of Mud in Agile Development – Can we Avoid Them -- 42

Draining the Swamp

You can escape from the
 “*Spaghetti Code Jungle*”

Indeed you can transform the landscape
 The key is not some magic bullet, but a
 long-term commitment to **architecture**,
 and to cultivating and refining “*quality*”
artifacts for your domain (Refactoring)!

Patterns of the best practices can help!!!

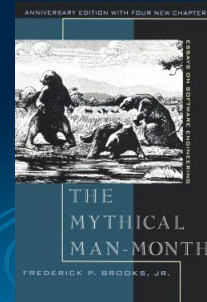
Big Balls of Mud in Agile Development – Can we Avoid Them -- 43

Silver Buckshot

There are no silver bullets
 ...Fred Brooks

But maybe some silver buckshot
 ...promising attacks

Good Design
 Frameworks
 Patterns
 Architecture
 Process/Organization
 Tools and Support
Good People ***



Big Balls of Mud in Agile Development – Can we Avoid Them -- 44

Mud is Here...

*It isn't always bad!
It can be contained!
It can be cleaned up!*

Our code can be more habitable!

Big Balls of Mud in Agile Development – Can we Avoid Them -- 45

So There is Some Hope!!!

Testing (TDD), Refactoring, Regular
Feedback, Patterns, More Eyes, ...

Good People!!!

Continuous attention to technical excellence!

Retrospectives!

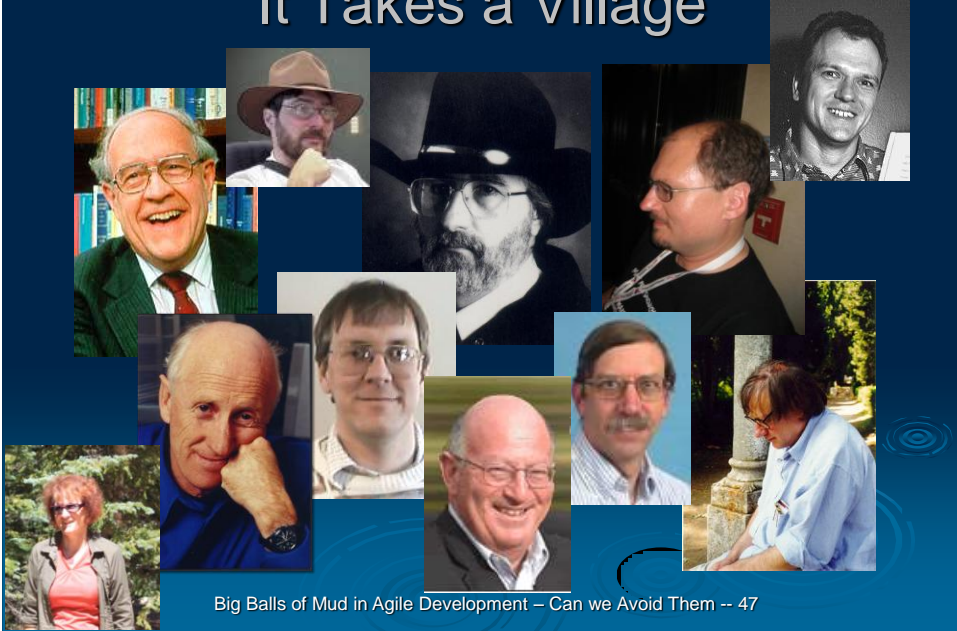
Face-To-Face conversation.

Motivated individuals with the *environment*
and *support* they need.

But, Maybe Mud is why we have Agile....

Big Balls of Mud in Agile Development – Can we Avoid Them -- 46

It Takes a Village



Thanks!!!



Big Balls of Mud in Agile Development – Can we Avoid Them -- 48