

# User Stories

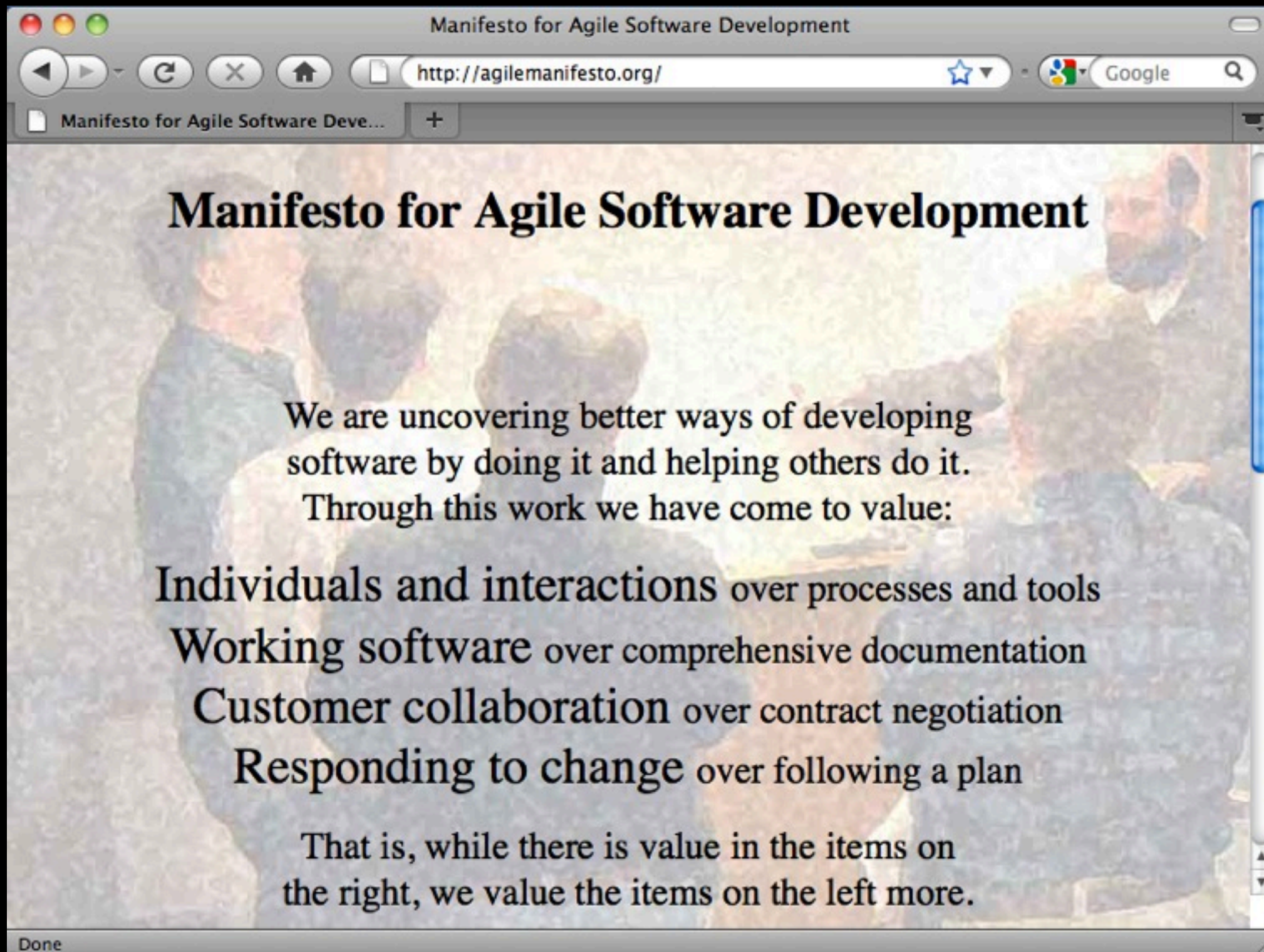
Agile and Lean Requirements

# Why User Stories?

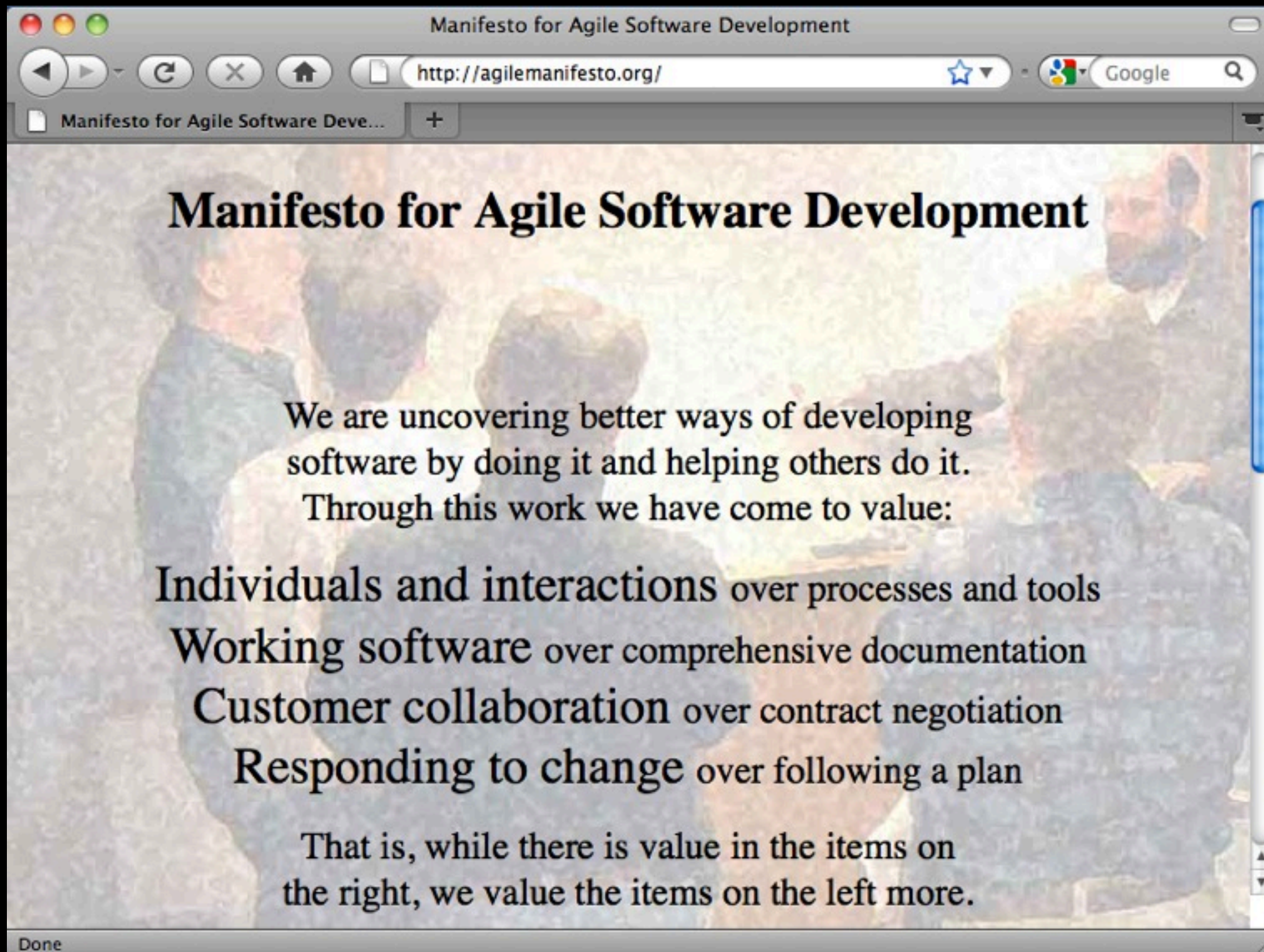
# Agility Reprise

...requirements churn is  
not a crisis.

# The Agile Manifesto







“over”, not “instead of”

# Underlying Assumptions of Agile



motivated, self-directed  
people

# cross-functional teams

# highly-available domain experts

# culture of testing





# Development-Initiated Agile Adoption

business still producing  
through phased analysis

usually functional  
specifications

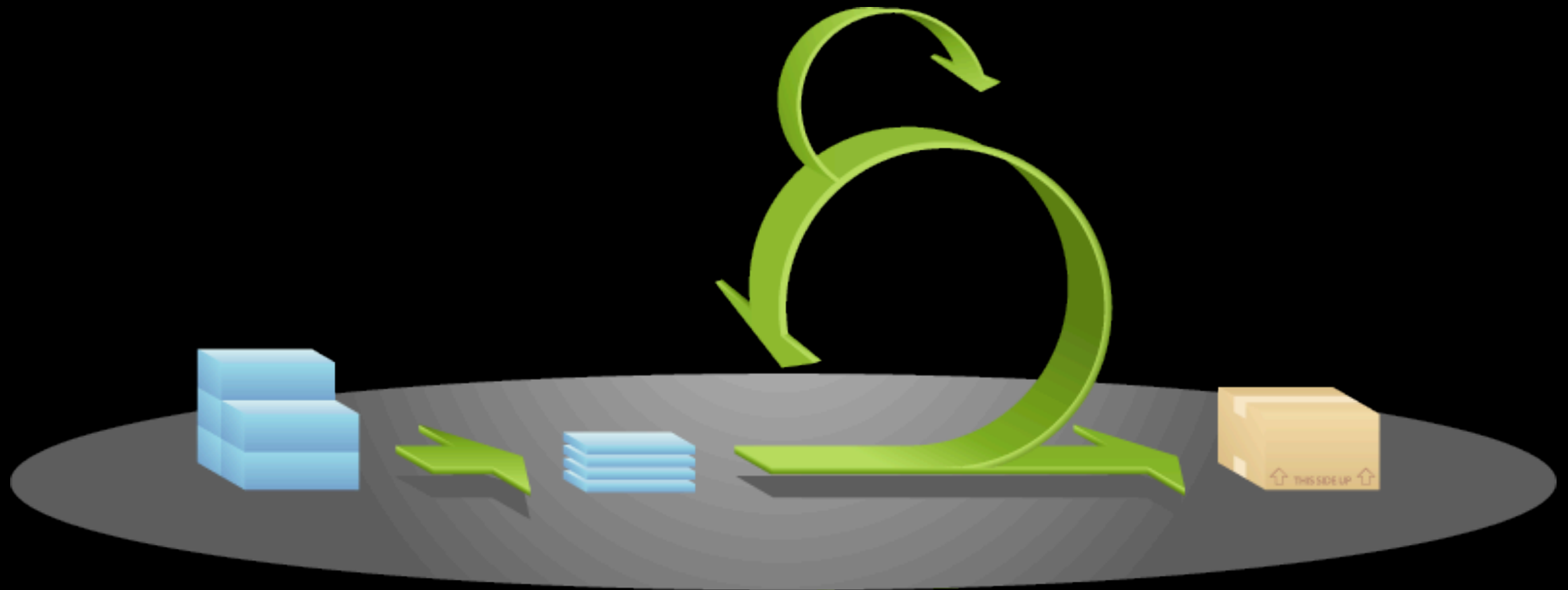
maybe use cases at best

# translational overhead



forfeits Agile's business  
value

creates challenges  
downstream

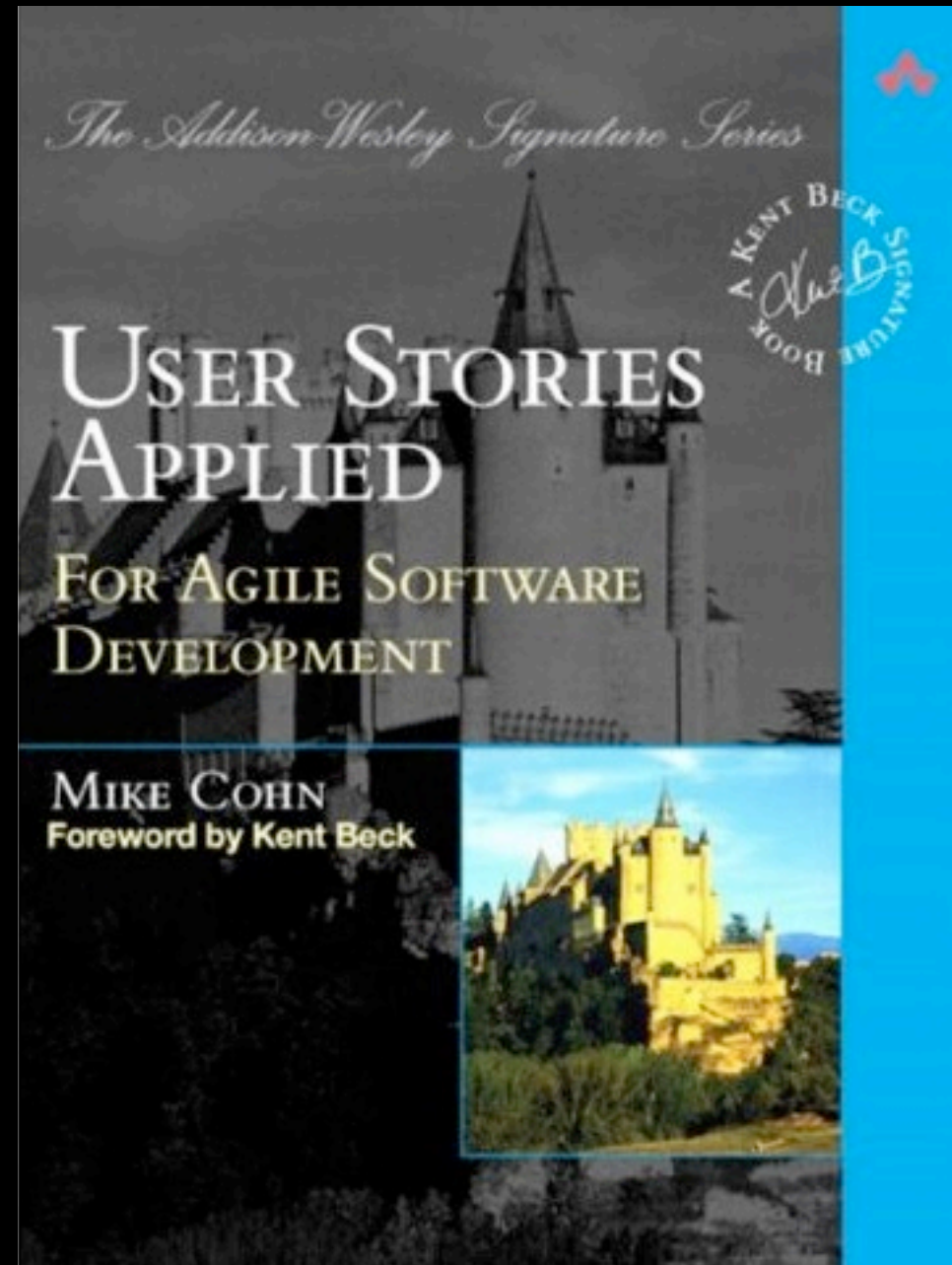


# Scrum



# What are User Stories?





the book to use

# placeholders for interaction

placeholder, not  
substitute

user stories are not use  
cases

smaller in scope



not permanent artifacts

too brief to stuff with  
UI requirements

# facilitate release and iteration planning

analysis catalyst, not  
analysis product

developed by the team



# What Should User Stories Look Like?

description of what's  
needed



# definition of success

dialog to clarify

as a \_\_\_\_\_ user,

I can \_\_\_\_\_,

so that \_\_\_\_\_.

# conditions of success

user stories are an art

# Bill Wake's INVEST acronym



independent

negotiable

valuable

estimatable

small

testable

But What About \_\_\_\_\_?

business won't  
cooperate



functional spec already  
written

team is not co-located

no UI for the code

# non-functional requirements

# technical debt

book examples are  
different





all things computed

Barry Hawkins

<http://alltc.com>

coaching/mentoring

agile software development

domain-driven design