

# Applying Agile Concepts in a Non-Agile Organization

*Geraldine Mongold*



I want to try Agile on my project but.....



I work in a really old-school industry.



I work for a state agency.

I work for a federal contractor.



# What is Agile really?

*More Important!*

*Less Important!*

Individuals and interactions

Processes and tools

Working software

Documentation

Responding to change

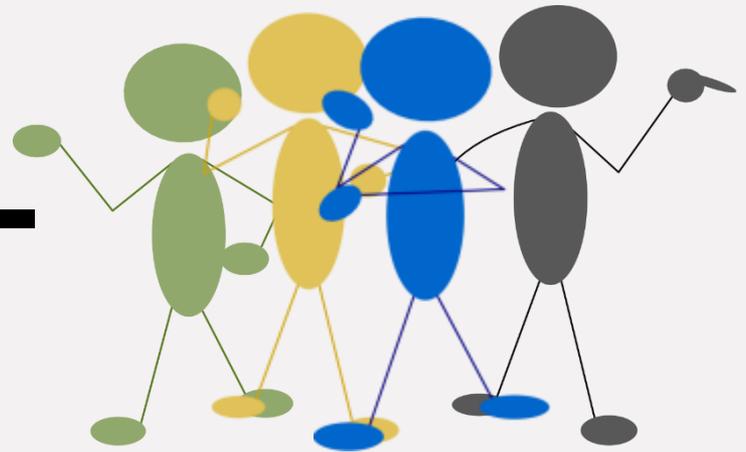
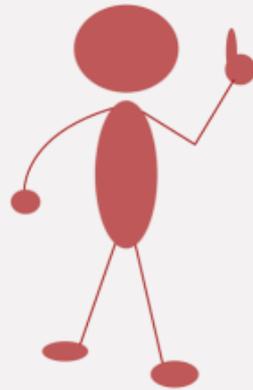
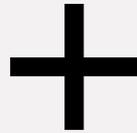
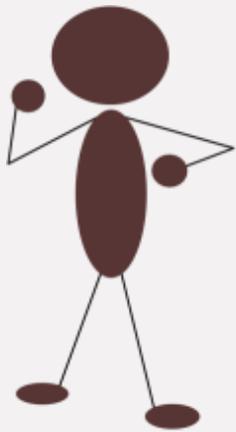
Following a plan

Collaborating with customers

Contract negotiation

# Flavors of Agile - SCRUM

Scrum Master – facilitator  
for the development  
team



Product owner  
(voice of the  
customer)

Development/delivery team (3-  
9 software developers and  
testers in a self-organizing  
team)

# Flavors of Agile - SCRUM

- Requirements are stated as *user stories* which are completed during defined chunks of time called sprints. A sprint is usually 2-3 weeks.



- Team Activities include a *daily scrum meeting* where each member briefly discusses yesterday's accomplishments, today's activities, and roadblocks.

- Each sprint concludes with a *sprint review & demo* of completed user stories.



# Flavors of Agile - Kanban

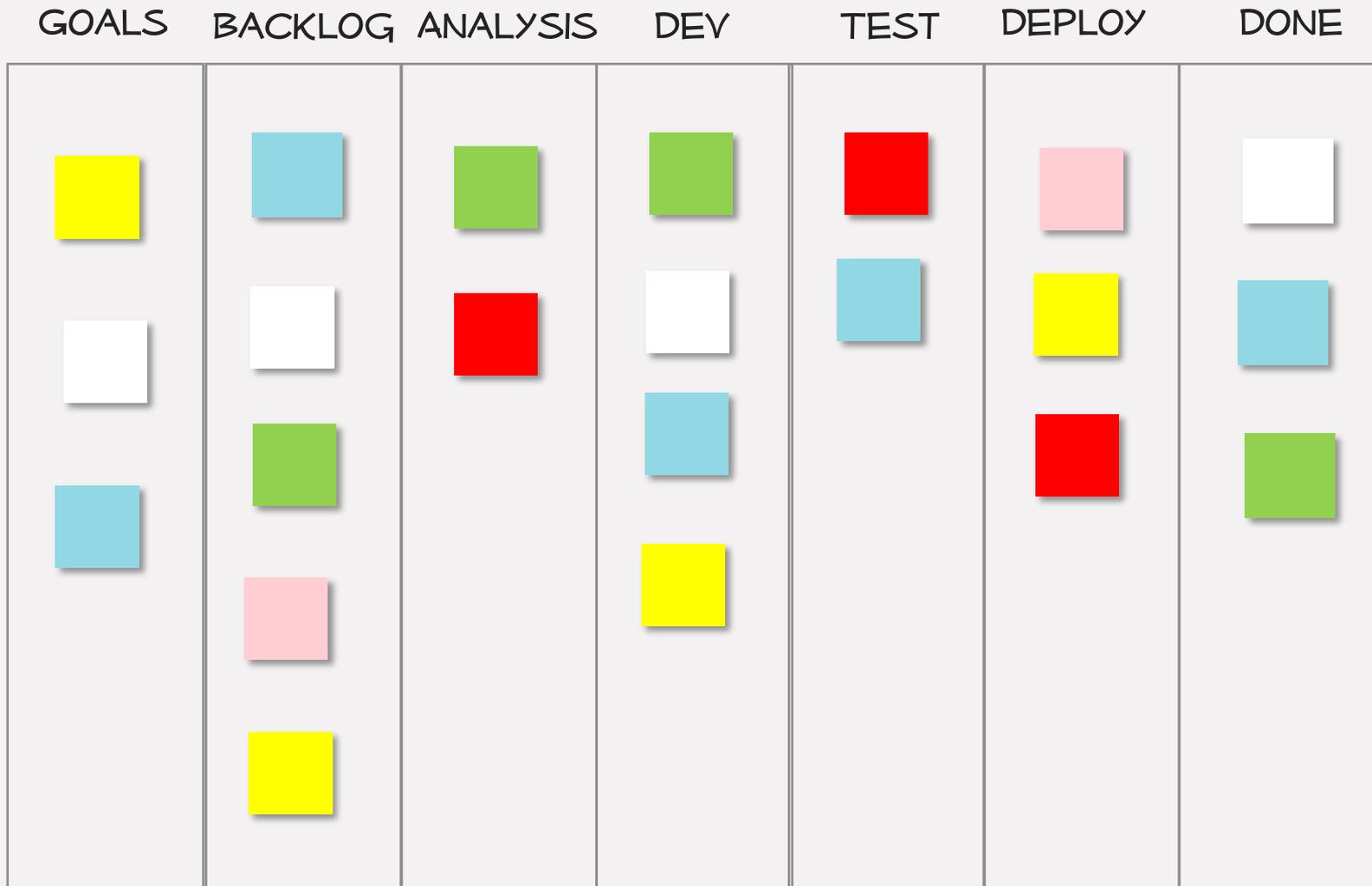
## *Differences between Kanban and SCRUM:*

- There are no time boxes in Kanban (*no sprints*)
- Kanban tasks are generally larger and fewer than *user stories*.
- There isn't an equivalent formality to *sprint review* although there may be periodic review events.
- There is no *team velocity* in Kanban — only average time for a full implementation is counted

看板

*Kanban  
means  
“signboard”  
in Japanese*

# Flavors of Agile - Kanban

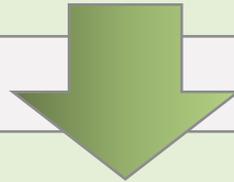


# Flavors of Agile – Behavior Driven Development

Behavior-driven development (BDD) focuses on the business behaviors the code is enabling - the *why* behind the code.



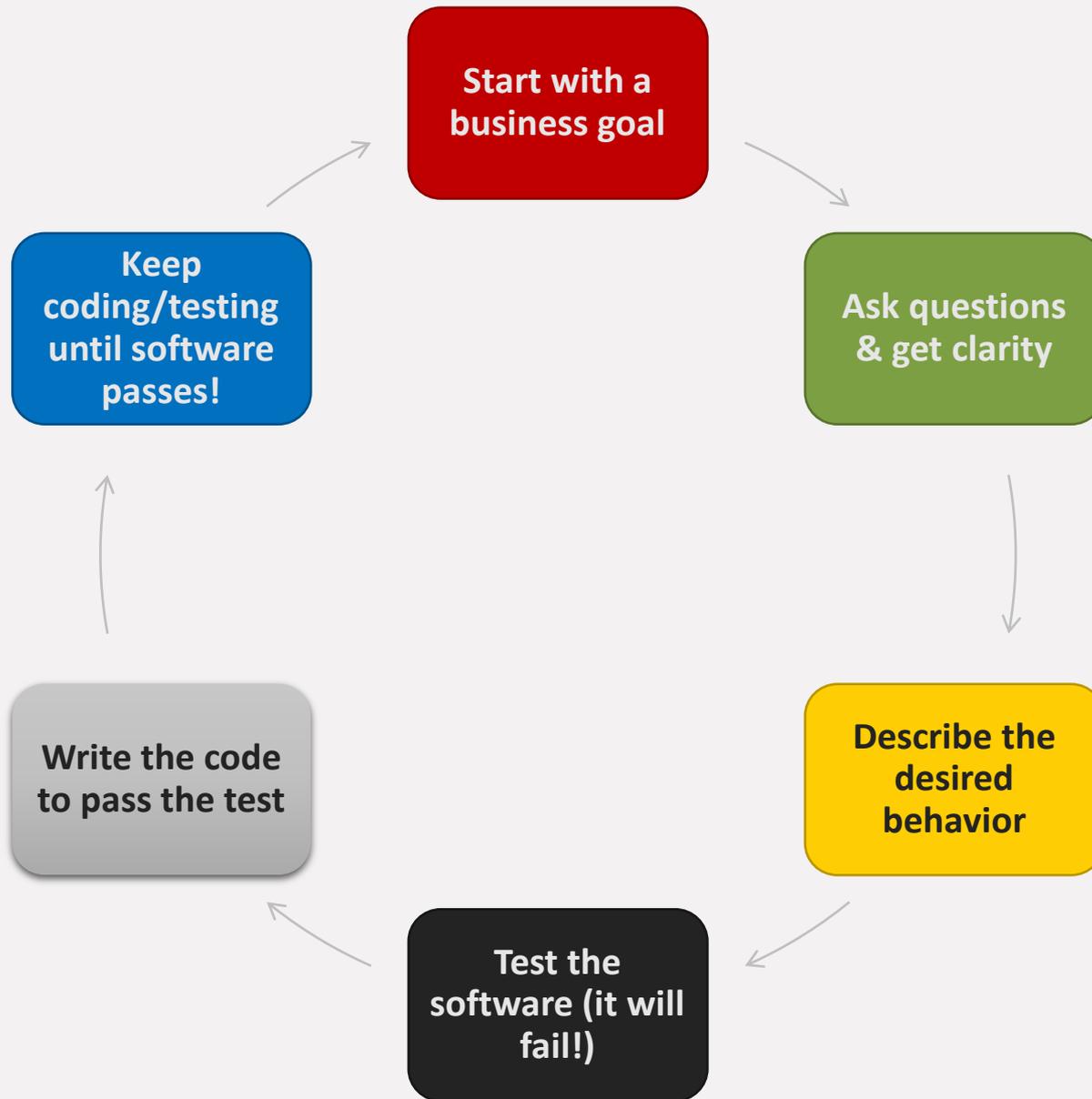
BDD artifacts are written:  
Given that \_\_\_\_, when \_\_\_\_, then \_\_\_\_.



## ***Example:***

Given that the user is logged into the account,  
When the user selects “Your Orders”  
Then the user will view all of their orders for the past 6 months,  
starting with the most recent.

# Flavors of Agile – Behavior Driven Development



# What should I do first?



- It depends on what's causing you the most pain!
- What are your top 3 software development problems?
- Remember the 80/20 rule – changing a few things can have a big impact.

# How can I fix this pain?

I don't know what everyone is working on. It seems to be taking too long to get tasks completed. What's up?



[www.pictofigo.com](http://www.pictofigo.com)



Implement team stand-up meetings to improve communication and accountability.

# How can I fix this pain?

Our requirements documents are so big, they take a long time to write and nobody ever reviews them completely.



Switch to an Agile Requirements Document – eliminate non-value-added boilerplate content & reduce the number of approvers for project documentation.

# How can I fix this pain?

There never seems to be enough time for testing. Defects get missed, resulting in rework and angry stakeholders.



Estimates for each feature include testing time. “Done” means testing is complete. Testing includes stakeholder review and acceptance of completed requirements.

# How can I fix this pain?

Requirements prioritization is a joke. Stakeholders know that if it isn't a must-have, it won't get built. So guess what? Everything is a must-have!



If you're still using MoSCoW, stop! Stack rank business requirements with 1 being the highest priority.

# How can I fix this pain?

We have to estimate the entire project up-front, so teams pad the estimate because they don't know what surprises might come along later.



Break up your requirements into chunks that are roughly equal (features) Use a default sizing for your features & elaborate as the project progresses.

# How can I fix this pain?

Our change management process is really heavy, so people either go around it and make unauthorized changes, or needed changes just don't get made.



Break the proposed change up into features and apply default sizing. How does this change prioritize against the other stack-ranked requirements?

# Being Agile – the marshmallow challenge

## Build the Tallest Freestanding Structure



20 sticks of spaghetti + one yard tape + one yard string + one marshmallow

# Being Agile – the marshmallow challenge

## The Rules

1. Build the Tallest ***Freestanding*** Structure
2. The ***Entire*** Marshmallow Must be on Top
3. Use as Much or as Little of the Kit as you need
4. Break up the Spaghetti, String, Tape
5. You have 18 Minutes to Finish

# Being Agile – the marshmallow challenge

Results from  
Product Camp  
March 2015...

26 inches!



# Being Agile – the marshmallow challenge

What approach did your team take?

Did you design first and then build once?

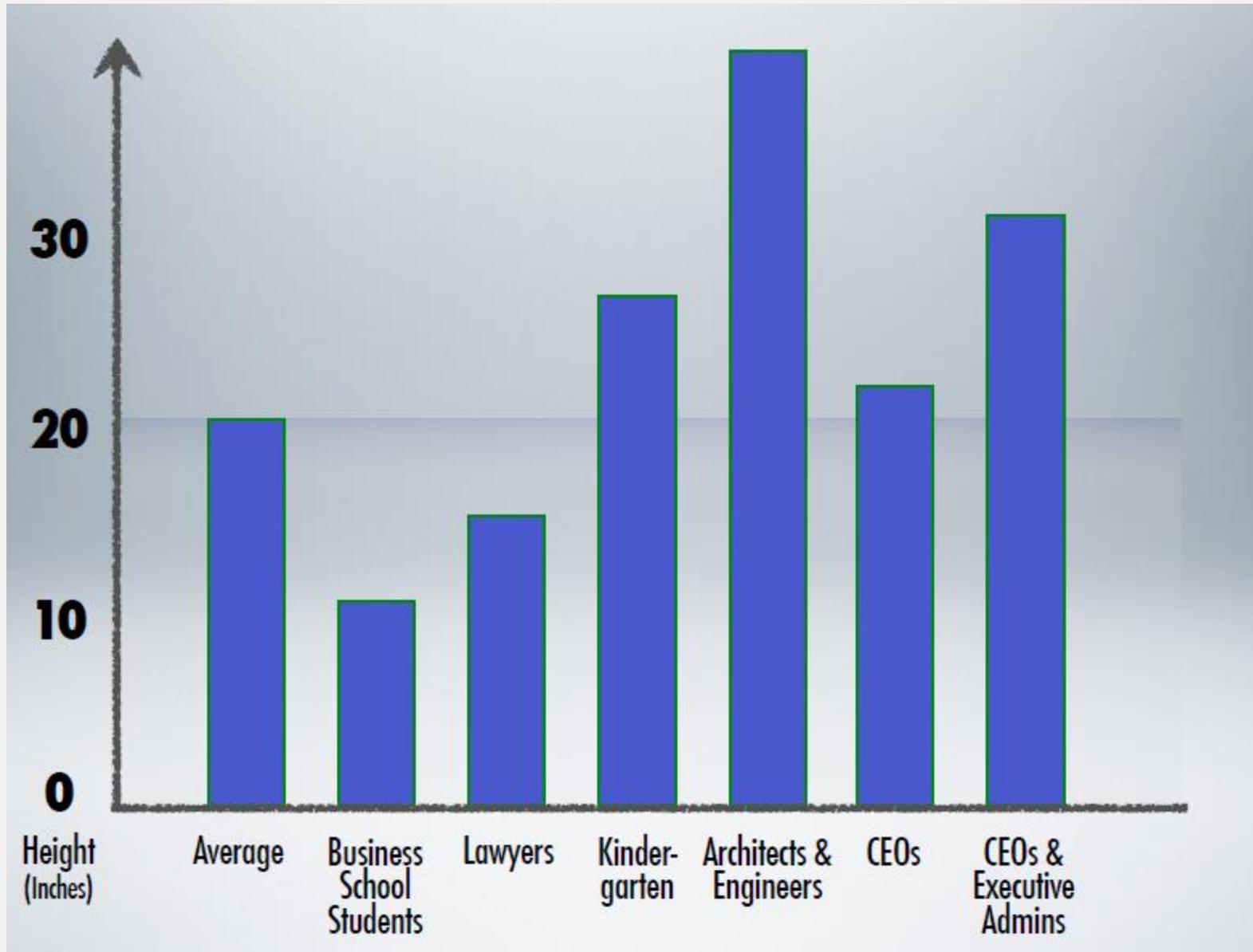
Or

Did you immediately start tinkering?

Did you have to tear down and rebuild?



# Being Agile – the marshmallow challenge



# Being Agile – the marshmallow challenge



# Being Agile - Summary

Agile is a **philosophy**, not a methodology.

Agile concepts can be applied in **any organization**.

Identify your **pain** points.

Introduce Agile **processes** to mitigate the pain.

Accept that you will **fail**.

Channel your inner **kindergartener**, adjust course, and carry on!