

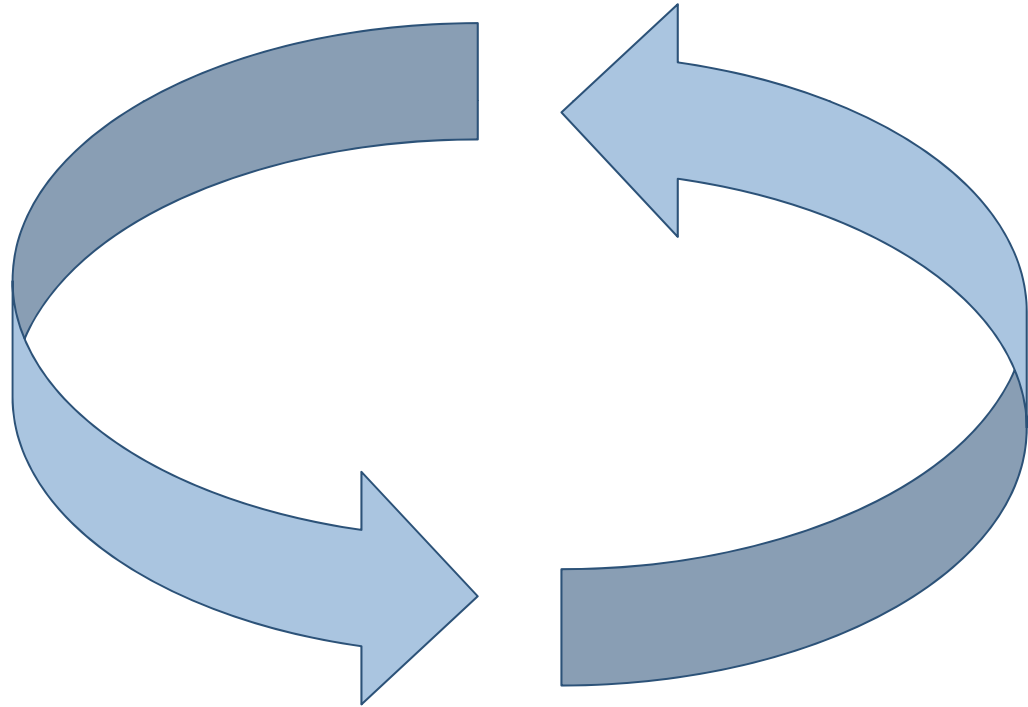
# **Applying Agile Game Development Techniques to eLearning**

Antonio Guadagno  
Matthew Tang

# Agile Defined

Any development methodology that is

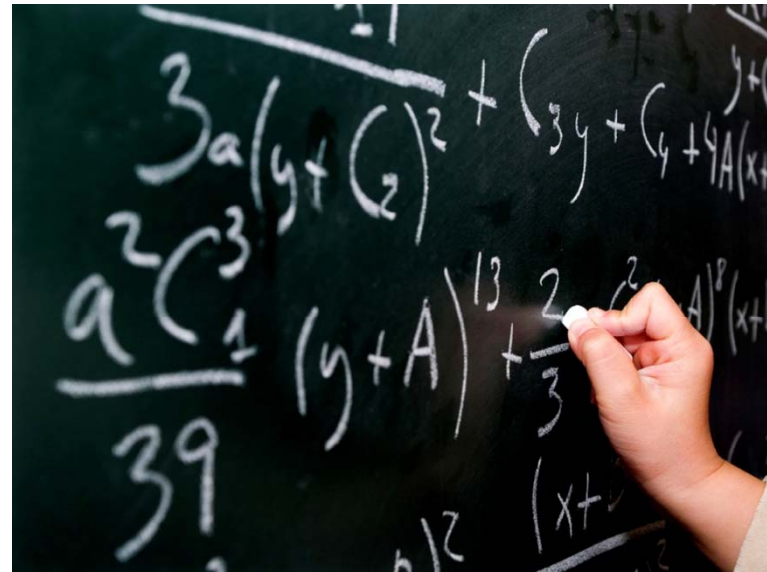
- Iterative
- Flexible



# Why Agile?

Technology enabling more complex solutions

- Games
- Simulations
- Branching Scenarios



# The Old Way of Development

- 1) Receive Storyboard
- 2) Build templates
- 3) Develop
- 4) Review course (1<sup>st</sup> draft)
- 5) Review course (2<sup>nd</sup> draft)
- 6) Review course (Final)



Who uses this process now and how does it work for you?

Type your answer in the chat

This method still works!

# New Solutions = New Challenges

Complex solutions created new challenges

- Longer development
- Higher cost
- More risk
- New expectations



# Finding Answers

Where may processes and tools to deal with these new challenges already exist?

- Software Engineering
- Electronic Game Industry



This led us to agile development methods

# Why Agile?

- Quick Iterations
- Frequent Customer Reviews
- Scalable
- Established Processes
- Small Pieces, Easy to Fix
- Can Always Deploy
- Timebox



# Developer's Perspective

- Iterative development
- Testing
- Estimation
- Develop for the future

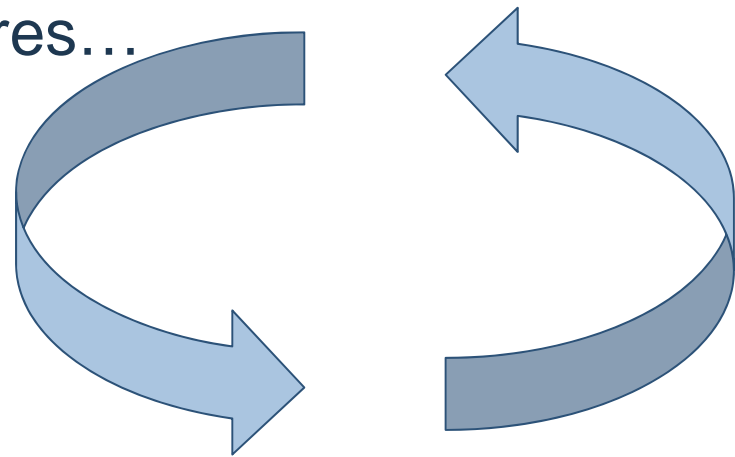


# Iterative Development

- Prototyping and iteration are common buzzwords in our industry and Agile
- This Agile method is best for new solutions
- Use both when trying something new
  - The time added to the beginning will save more time at the end

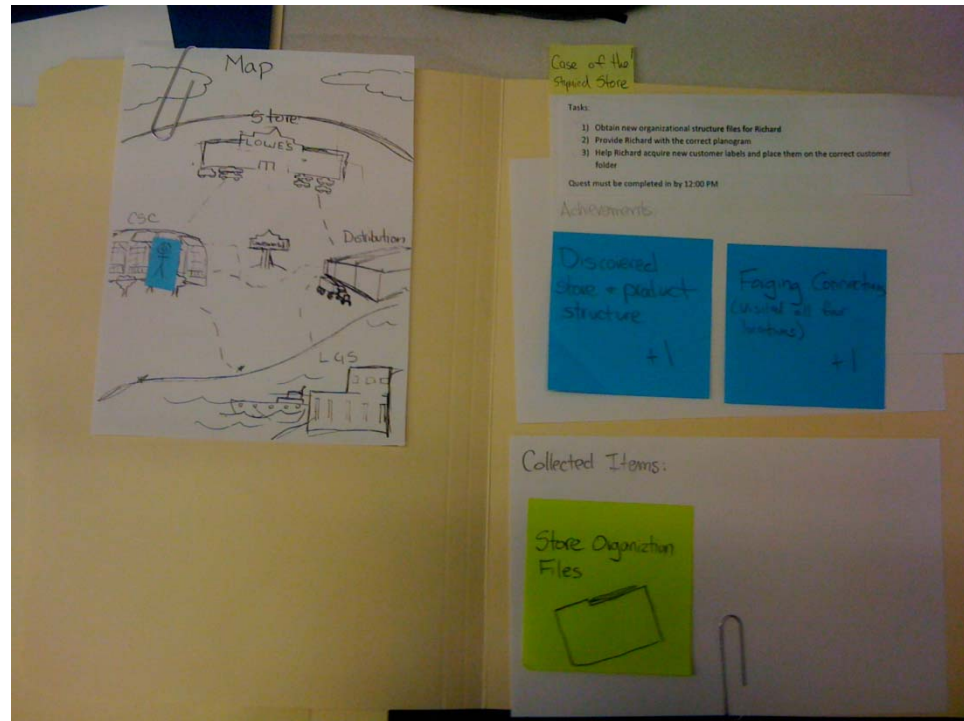
# Iterative Development

- Development walkthrough
  - Select a small subset of features
  - Develop this set
  - Review the set
  - Select the next set of features...



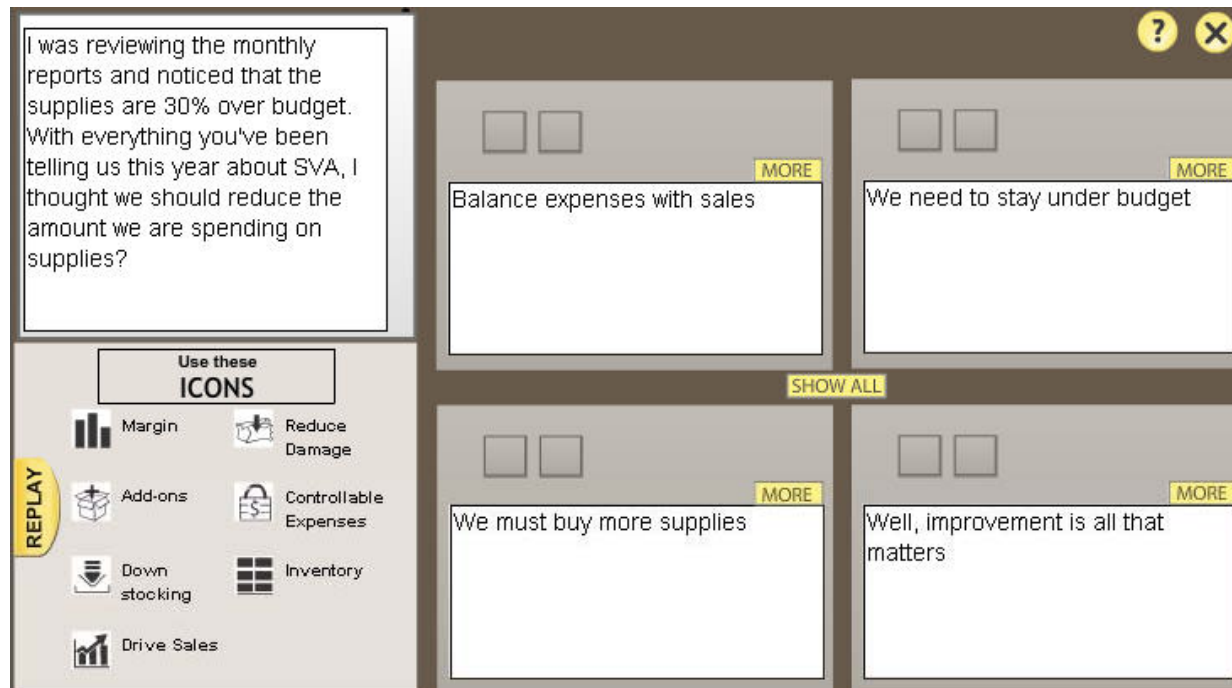
# Prototyping

- Start simple
  - Paper Prototypes
  - Extract Features
  
- Get creative!
  - Timers
  - Stop Watches
  - Have users “open the folder”



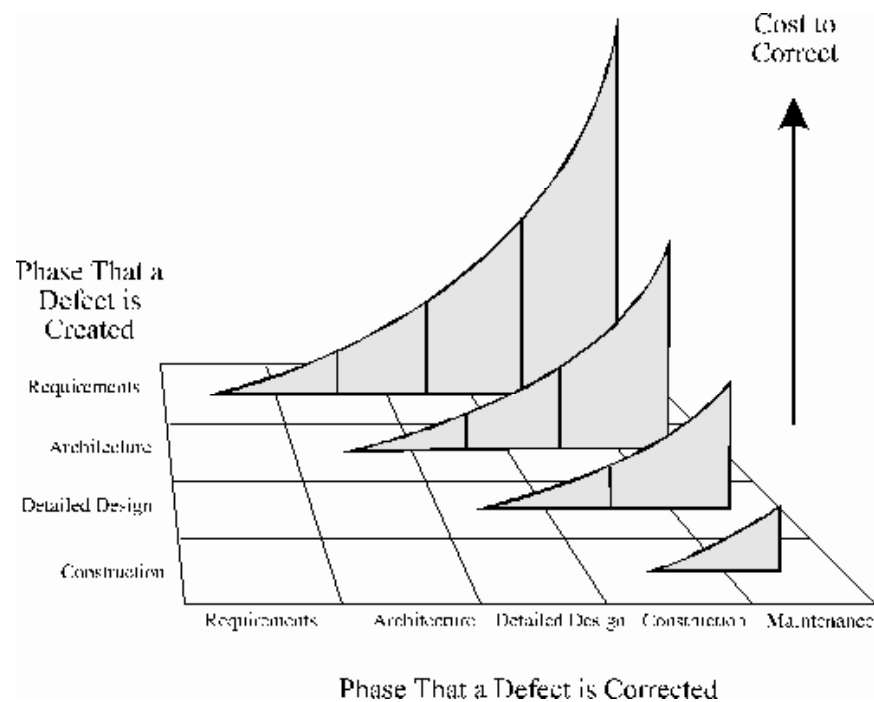
# Begin Play Testing Early

- We test on a weekly/biweekly basis
- We do not black box it anymore



# Doesn't Testing Add Time?

- Short answer is no
- It often adds time to the planned schedule
- It always saves time for actual delivery



# Use Who You Have!

- Coworkers/Friends will not give the needed constructive criticism
- Untrained “gamers” are not great testers
- Real end users, or the equivalent are best
  - If you have access, go to them or bring them to you
  - If not, find equivalent replacements (other retail employees, etc.)
  - Make fake content, if IP is an issue

# Testing Quick Tips

- Have at least one developer at every test
- Pretend you are someone else, not the developer
- WATCH! Don't just listen
  - What does the user hesitate on?
  - What do they spend too much time on?
  - What do they make confused looks about?
  - If they do any of these things on something not designed for that, it may be an issue to fix

# I Have Feedback, Now What?

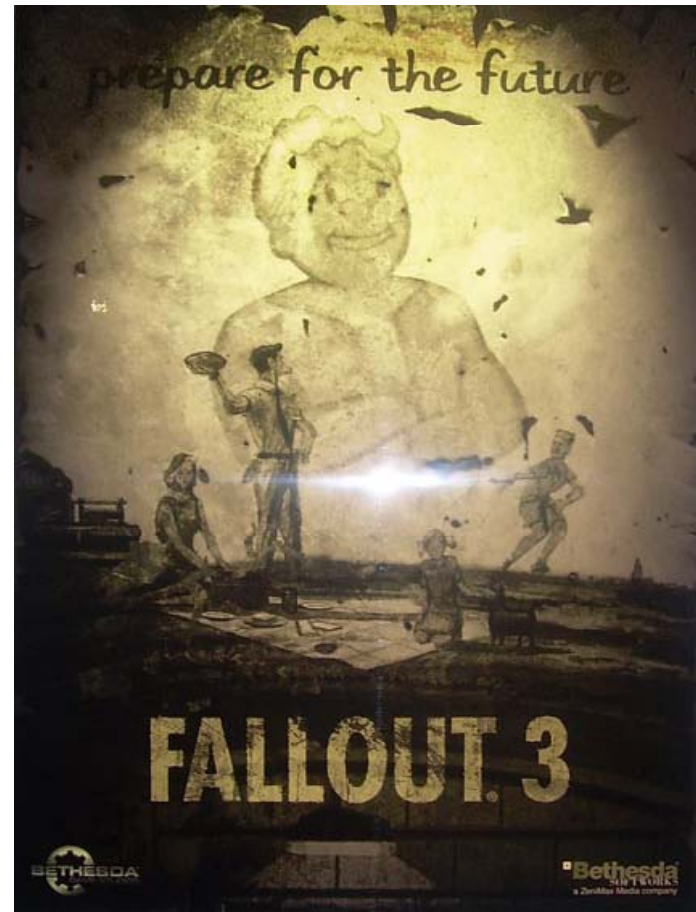
- Track feedback
- Review feedback with others on the team
- Rate feedback
- Add the feedback you choose to take action on to your course development estimation

# Quick Detour Through Estimation

- Doesn't this take more time?
- Our Estimation Tool

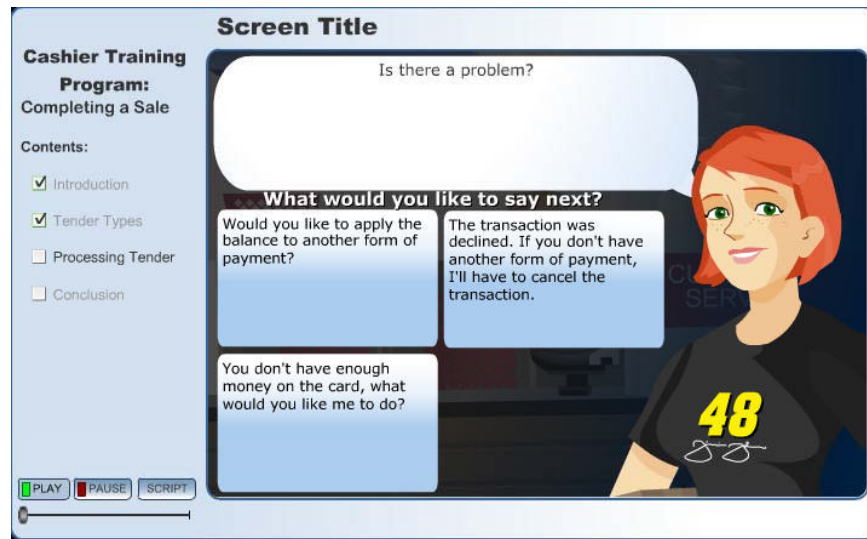
# Develop For the Future

- XML



# Develop For the Future

- XML
- Use Classes



# Questions?

- Matthew Tang - [matthew.tang@lowes.com](mailto:matthew.tang@lowes.com)
- Antonio Guadagno – [antonio.p.guadagno@lowes.com](mailto:antonio.p.guadagno@lowes.com)