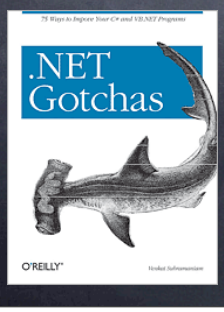
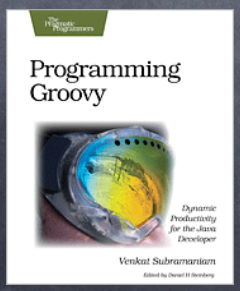
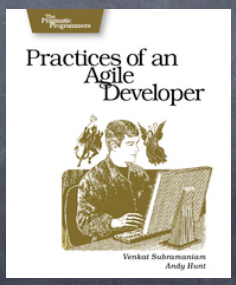
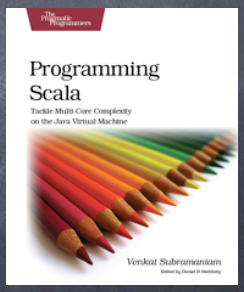
# *fallacies of agile development*

Venkat Subramaniam
email:venkats@AgileDeveloper.com
twitter: venkat_s




In 3 words or less, what's Agile Development?

# What Comes To my Mind

Discipline

Hard work

Fun

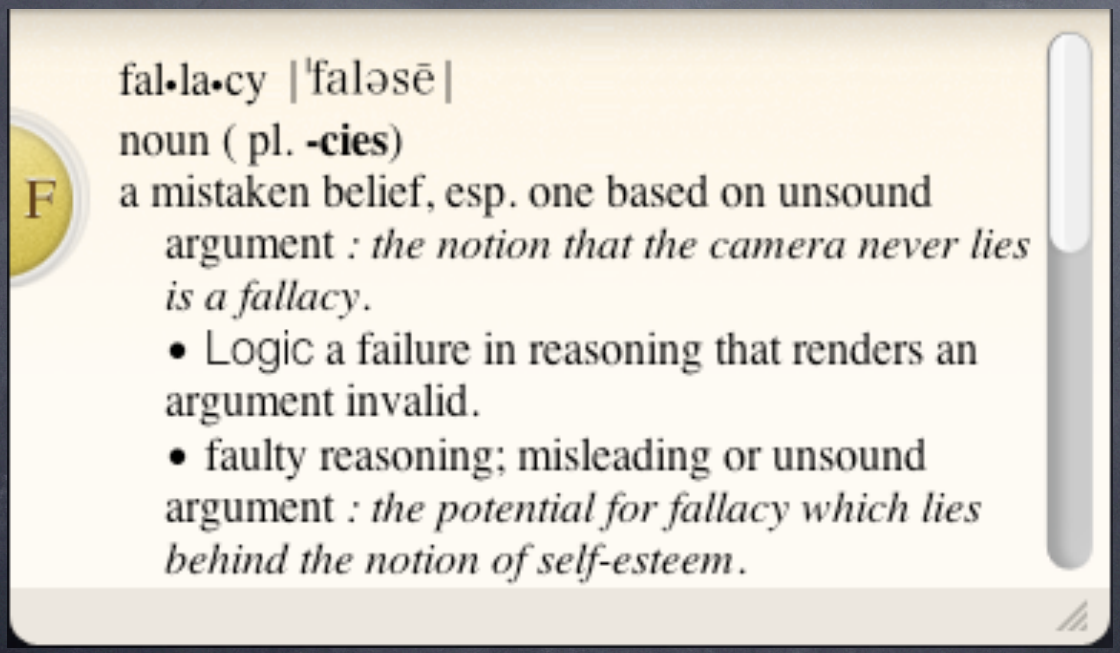
Continuous, not Episodic

Customer Participation

Relevant Working Software

Feedback Driven Development





In this presentation I'll discuss a few Fallacies I've come across over the past couple of years

There are other fallacies too, so the list is not comprehensive in any way



If you only care for speed, the result may be

You can gain speed here

# Slowdown; Enjoy the Ride



Can you respond to Change?

Can you respond to Feedback?

Is change affordable?

Your Practices should promote sustainability



# Fallacy: Testing Interrupts Development

Words I heard recently: "Our Scrum Master defends interruptions...

...he does not let testers interrupt our programmers"




What's your definition of "done"?

coded?

Acceptable?

Potentially Releasable?



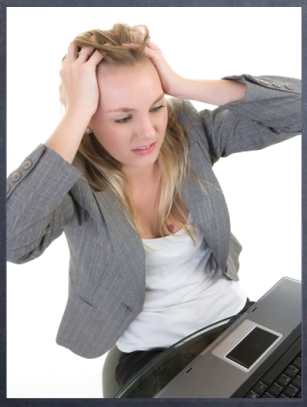
Testing is essential to make development sustainable

# Feedback comes in different forms

Programmer Unit Tests

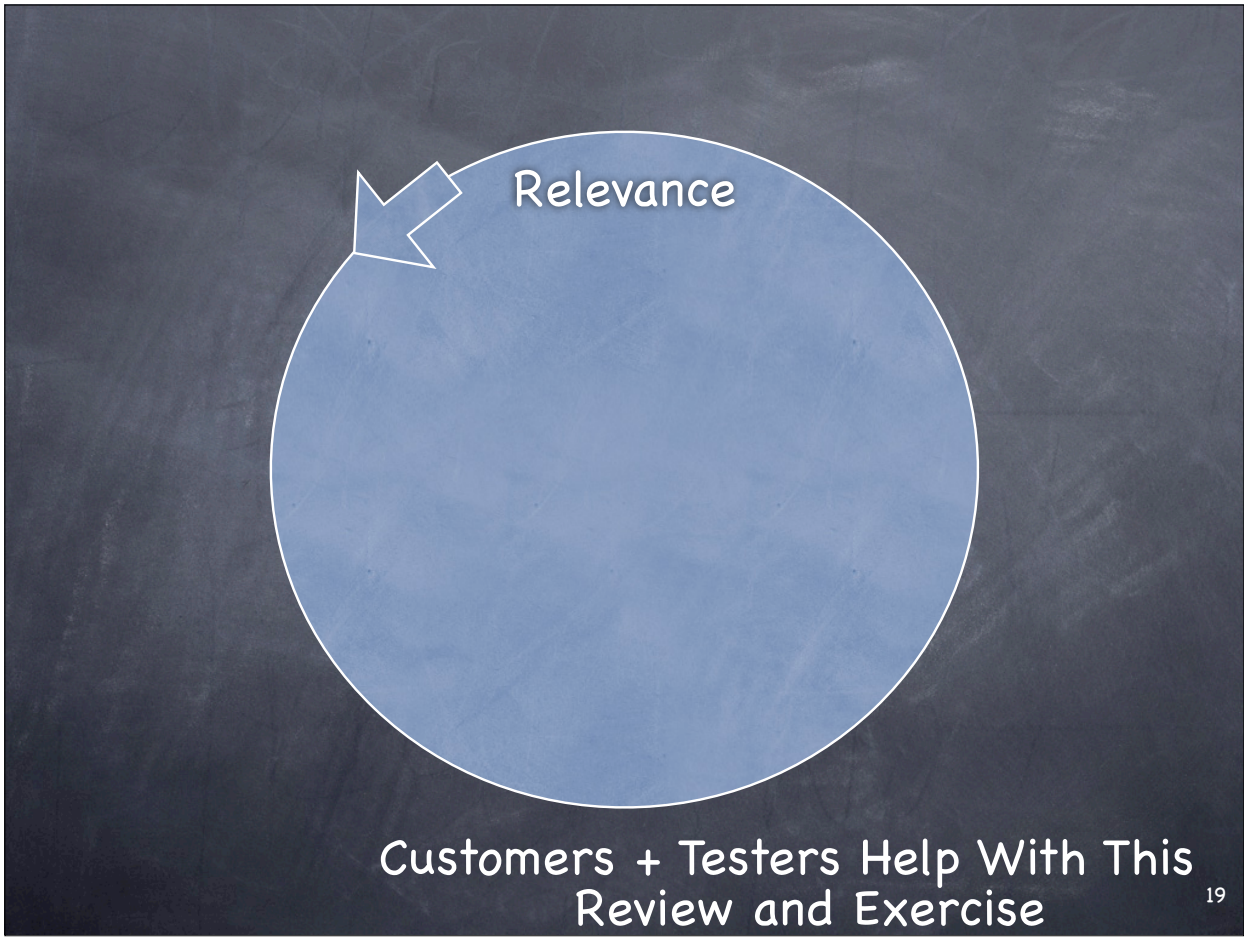
Automated Acceptance Tests + Manual Tests

Customer Interaction + Review & Exercise



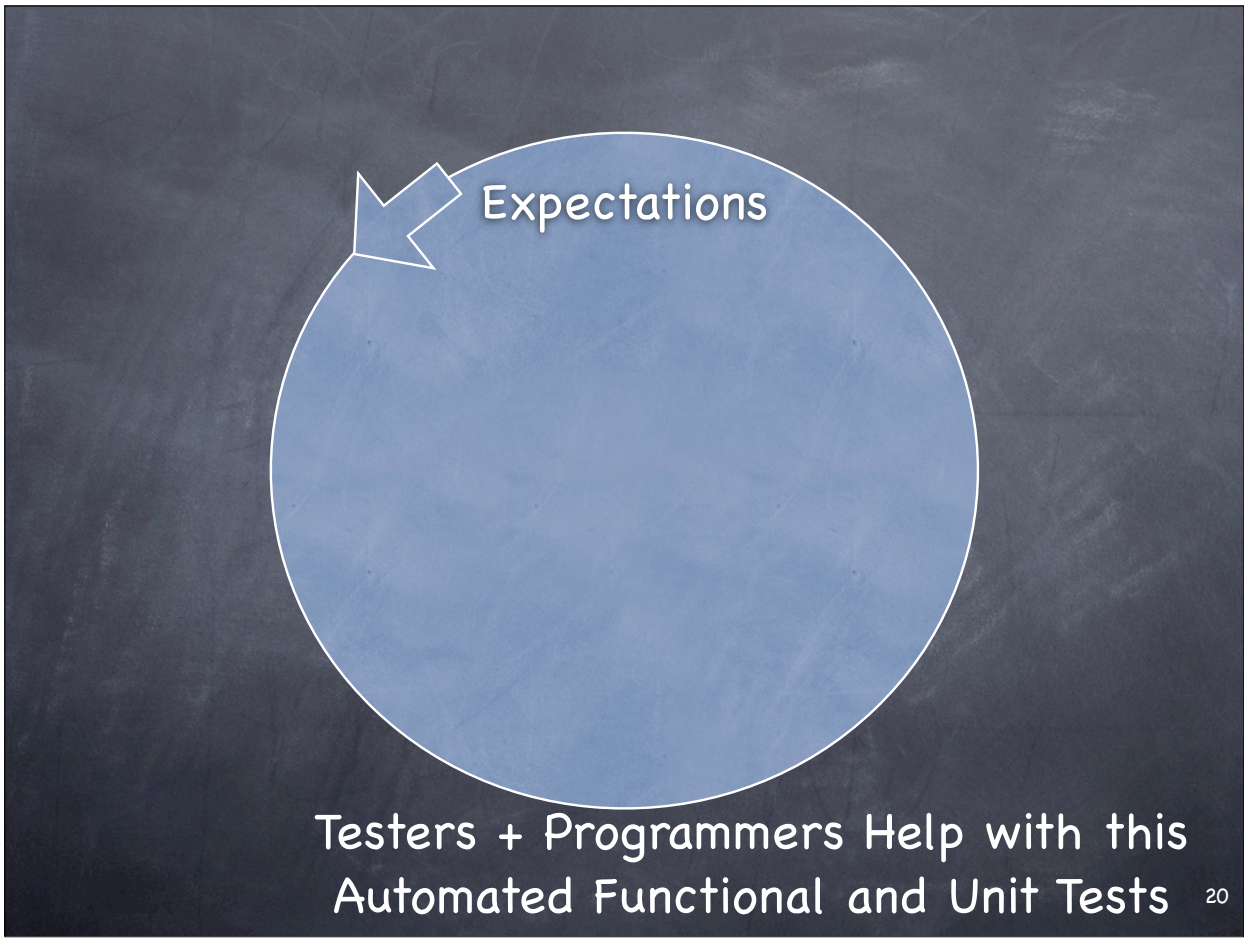


# Which of those is important?




Customers + Testers Help With This Review and Exercise

19




Testers + Programmers Help with this Automated Functional and Unit Tests

20

What if you can't have both?



But software exhibits Heisenberg Effect

You Get Feedback

Your Team Responds to Change



Customers and Testers Find Parts of the App Broken Now

# Team Responds with Quick Fix

Other Parts Broken...



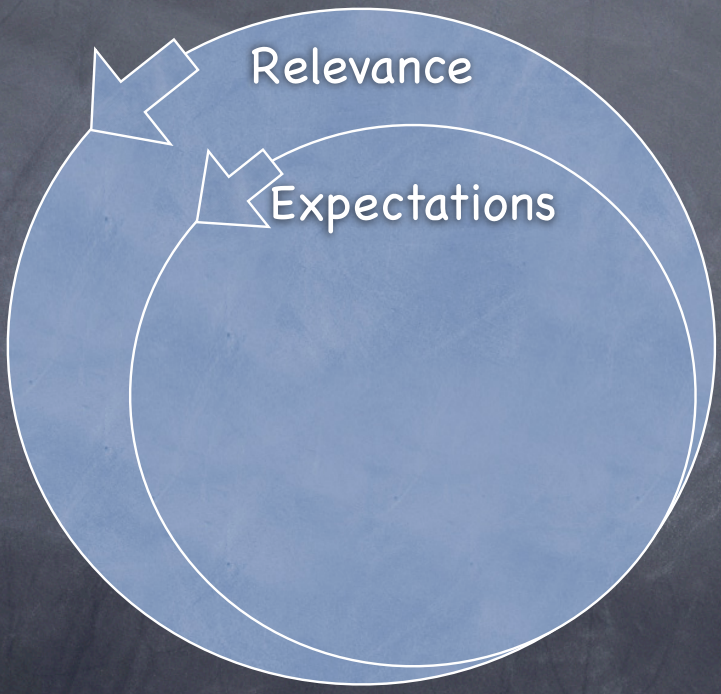
# What have you built?

Whack-A-Mole System!!



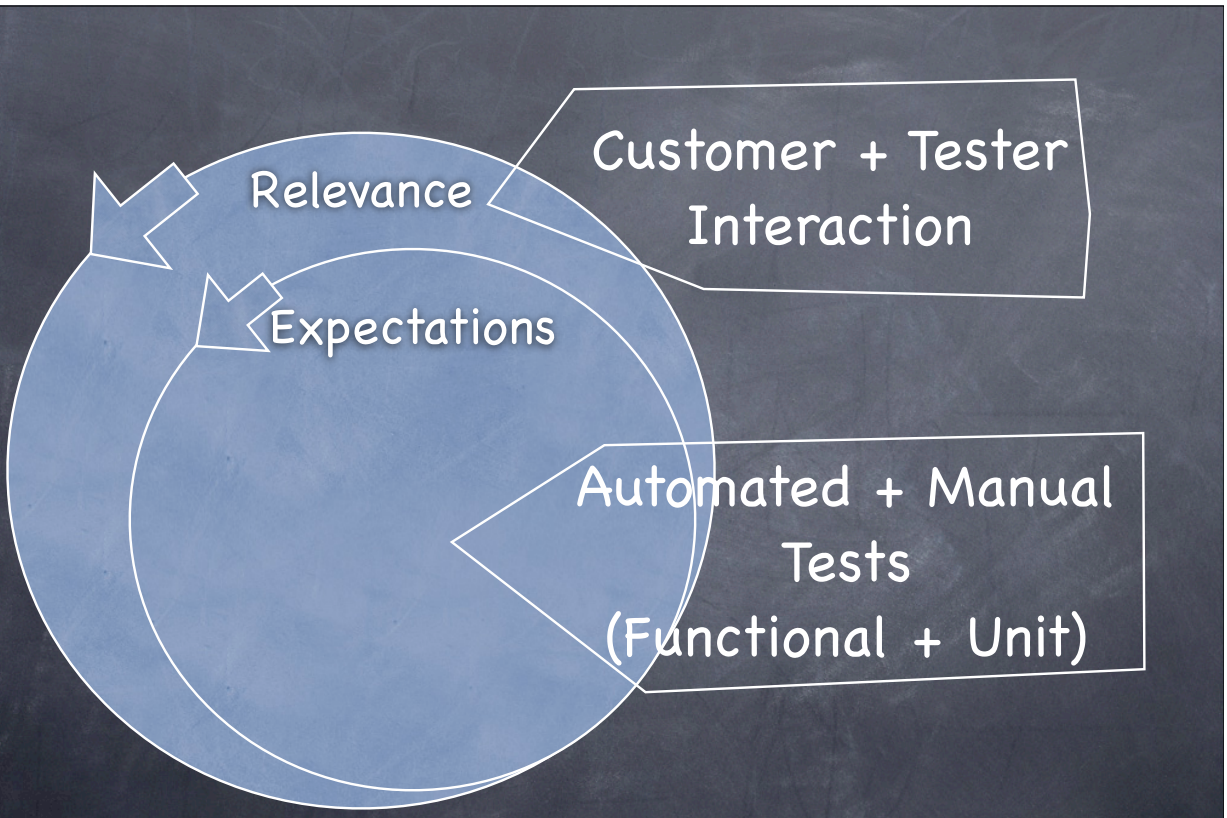
Not What You Set Out For

# Can' Have One Without the Other




Inner circle sustains the Outer





## Fallacy: Self Managed==No Leadership

Self Managed is Good

What does that Mean?

Do not assign tasks

Do not Micromanage

Let Team Interact

Promote Openness

Team is Cross–functional

# Dictators are bad

They make arbitrary Decisions

They exercise excessive power



# Every Project Needs a...


Benevolent Dictator

Someone who can be approached with questions

who can call the shots and get everyone moving forward




Fallacy: Collaboration == Everyone Involved on all Aspects

Empowerment is good

Collaboration is key

That does not mean...
everyone attends all meetings...
... involved in all decisions



A manager was keen on empowering...

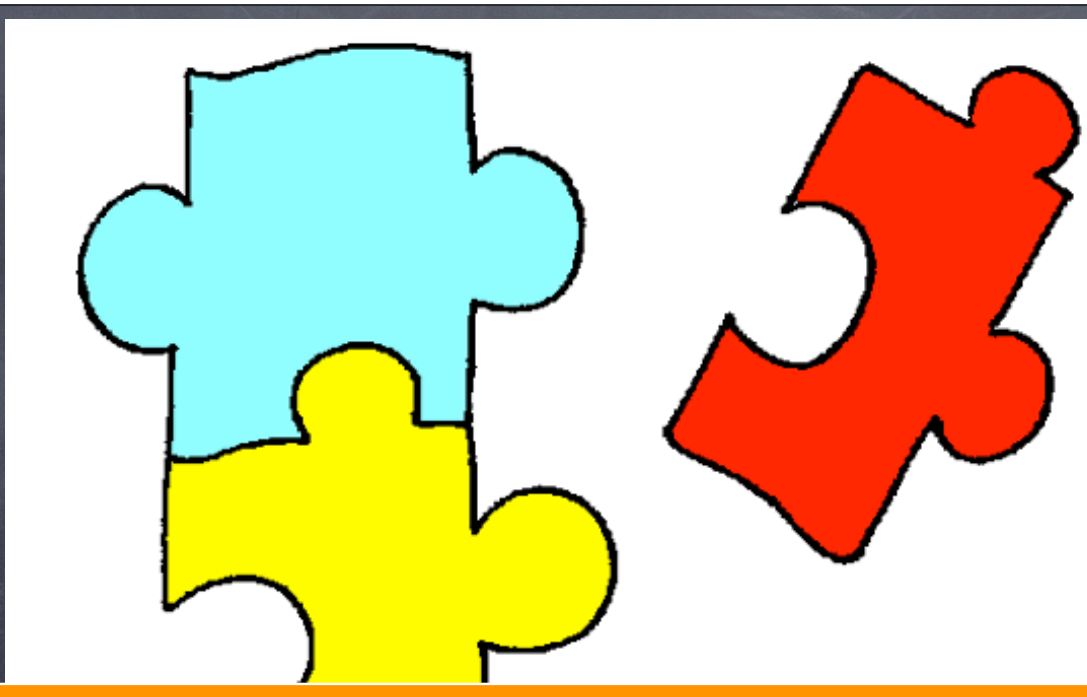
the team found it overwhelming, frustrating, and poor use of their time

Pair up Developers, Testers,...

Each pair solves problems
They pull customers when they need

Rotate pairs frequently
Avoids truck factor
Improves Competency
Promotes continuous review





Fallacy: Agile Will Fit Into Current Organization

What's Your Org Structure?

Testers not part of team?

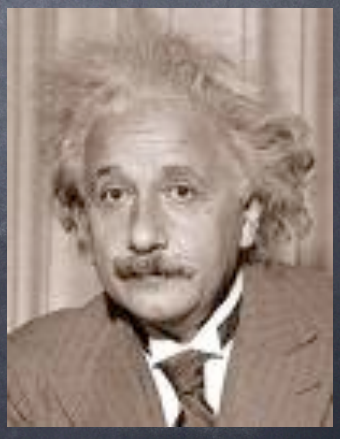
Can't ask for customers?

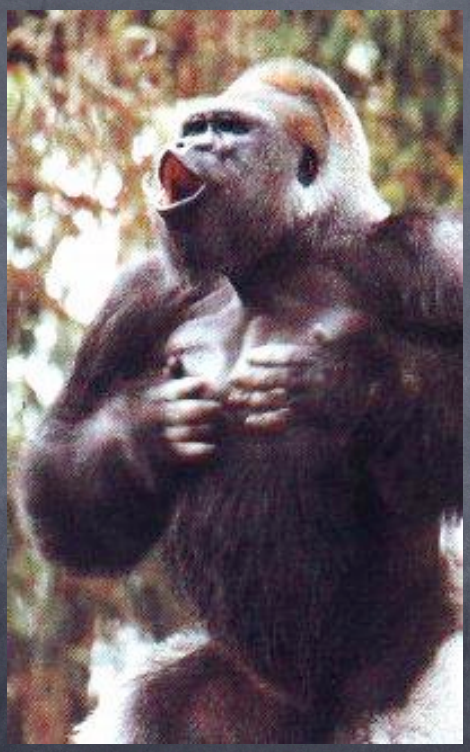
Reject things easily as Ideal?

How's change accepted?
Question constraints



"Insanity: Doing the same thing over and over again, and expecting different results"







If you're not willing to change
... don't expect to see change

Negotiate ways that promote success

Adapt, reevaluate, reinvigorate



Fallacy: Can't Break Builds



Breaking build
not a crime

It happens

Incompatibility of libraries

Idiosyncrasies

Slow integration tests

...

If you fear broken build, you resist frequent checkin

That's even worse
It hides the problem
postpones it



Do not penalize for breaking

But, don't tolerate build left broken —set a time limit

# Fallacy: Retrospection is enough to improve process



Retrospection is important

Brings Team together

Can voice concerns

Gives a breather

You can figure ways to improve

But,...



Not the best time to collect all likes and dislikes

Iteration pain often forgotten during retrospection

During iteration ask team to continuously keep a log

| What's Working Well | Needs Improvement |
| --- | --- |
| … | … |
| … | … |
| … | … |

Update Continuously

Attend Critical Things Right away

Review at Retrospection



# Fallacy: No Need for Customers Since Programmers Know Domain

# Is Product Owner Available Daily?

Who represent your customers?



# A Burn Down I Found


# Perceived Progress Shattered by Product Owner Review

Daily Access to Product Owner

What if Product Owner Too Busy?

Have One or More Proxies to Product Owners—these I Consider as Customers



Having One or Two Customers is a good idea in any case

They serve as advisors to PO

They give feedback

They can interact with users

They can exercise App

(fill your favorite agile method here) will solve your problems?



Are you right Applying It?

Are you Applying It Right?

Ask your self:
How do you Feel?



It it does not feel right,...

It it does not give you results

It you are not succeeding...

# Thank You!

You can download examples and slides from
http://www.agiledeveloper.com - download