

# **Rediscovering XP - Extreme Programming**

**Resurgent? Or Retro ?**

**@Windholtz**

**[github.com/mwindholtz/presentations](https://github.com/mwindholtz/presentations)**

# eXtreme Programming

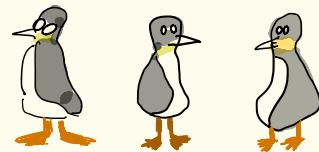
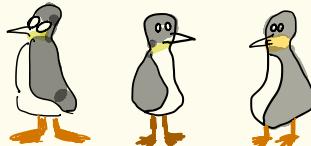
## Past - Present - Future

---

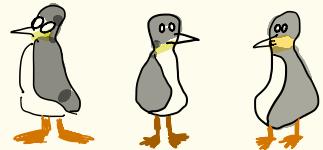
# Good presentation Practices

- 1) Start with Humor
- 2) Professional looking PowerPoint slides
- 3) Cute kitten photo at a surprising point

# Good presentation Practices



- 1) Start with a Humor
- 2) Professional looking PowerPoint slides
- 3) cute kitten photo at a surprising point

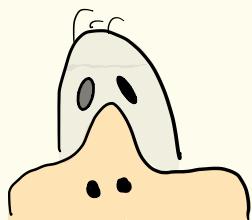
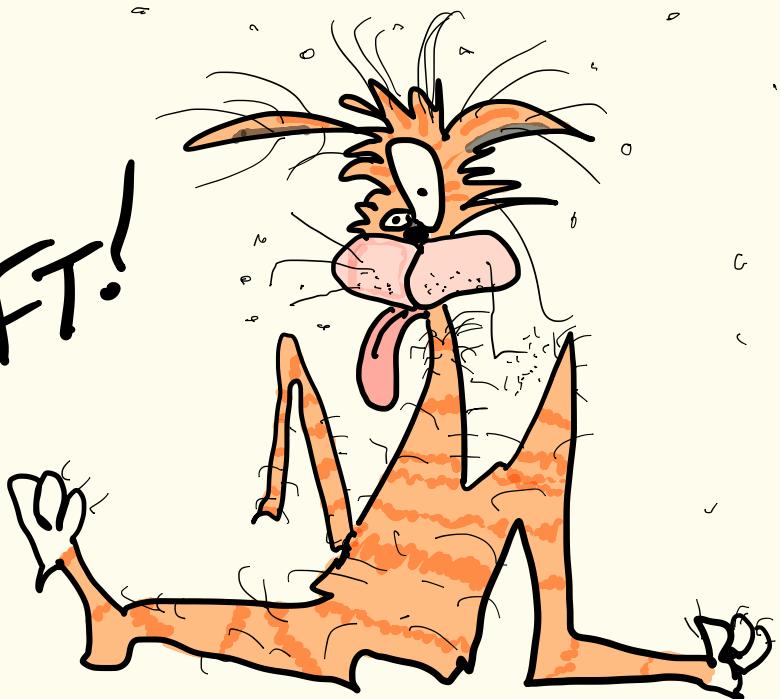


# Good presentation Practices

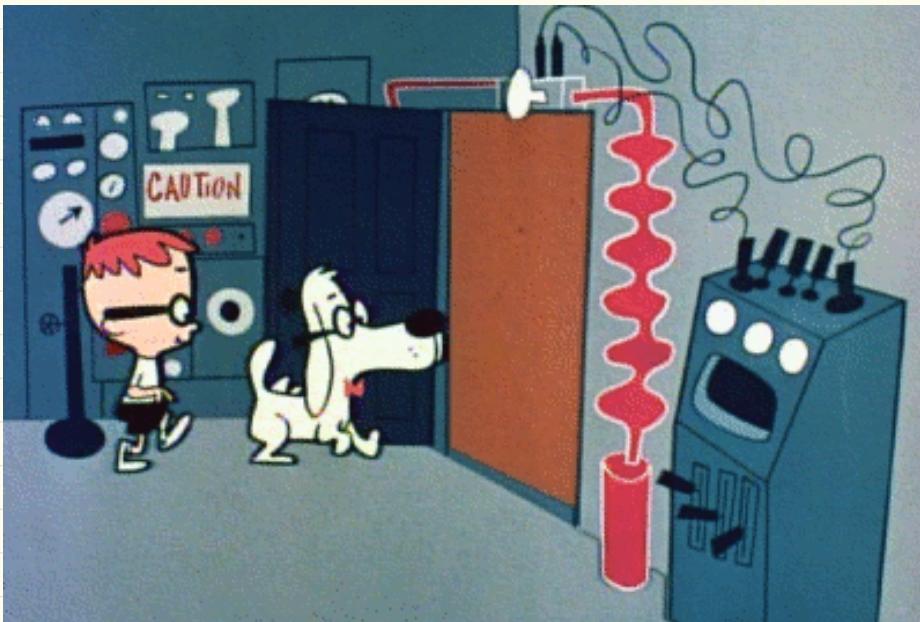
- 1) Start with a Humor ✓
- 2) Professional looking PowerPoint slides
- 3) cute kitten photo at a surprising point

# Bill the Kitten ?

ACK!  
THBBFT!



Sherman, set the Wayback machine to... 1992

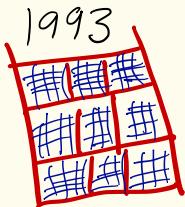


# Software Development

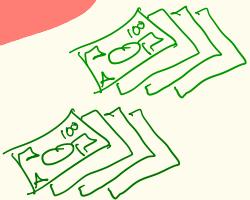
- ★ LARGE PROJECTS
- ★ YEARS LONG PROJECT PLANS
- ★ LOTS OF DEVELOPERS
- ★ UNDEFINED PROCESS

# Business Process Reengineering

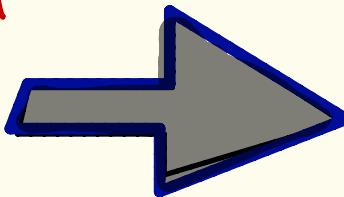
9 Months



\$30 Millions



9 MONTHS  
OF ANALYSIS

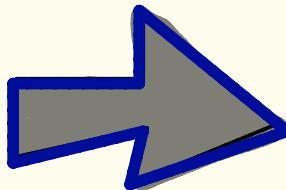
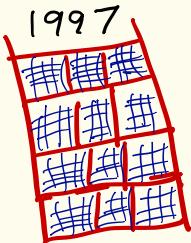


Canceled

# Department Of Defense

12 MONTHS

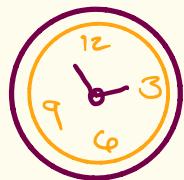
\$12 MILLION



REQ 2 M  
ANALYSIS 2 M  
DESIGN 3 M  
CODE 4 M  
TEST 1 M

Complete failure when deployed

12 MORE MONTHS CRUNCH-TIME  
3 SHIFTS  
\$48 MILLION



INSURANCE

3 YEAR

DEATH  
MARCH

\$120 MILLION

SEARCH  
ENGINE

CLIENT/SERVER

10 YEARS

CHAOS

FAILURE

PANIC

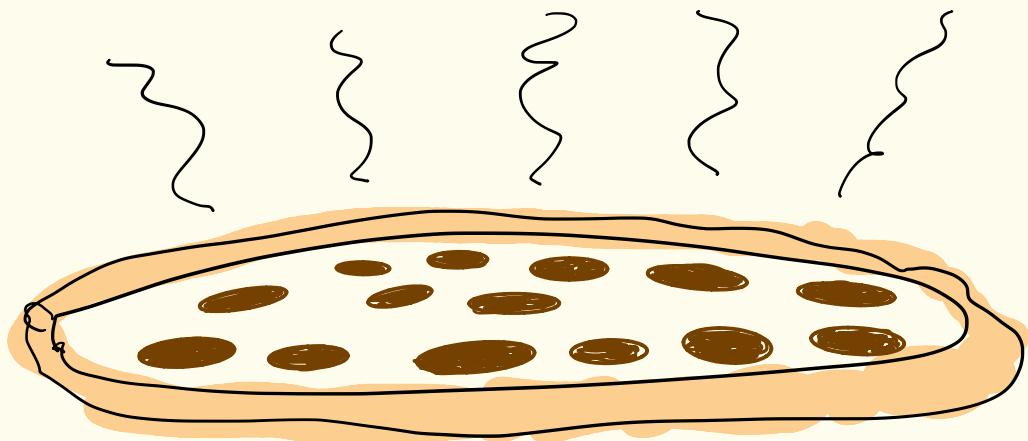
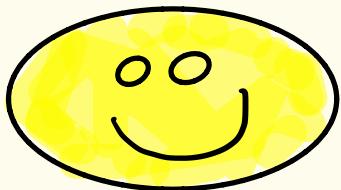
CAREER PROSPECTS

ME



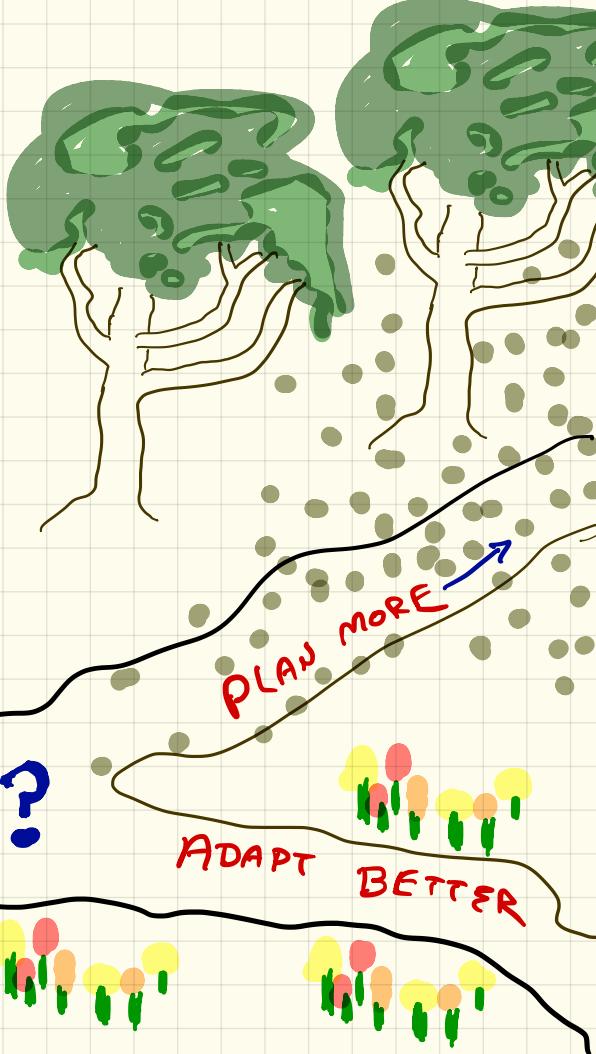
# Pizza !

Yum



# TWO PATHS

CHANGE IS SLOWING YOU DOWN

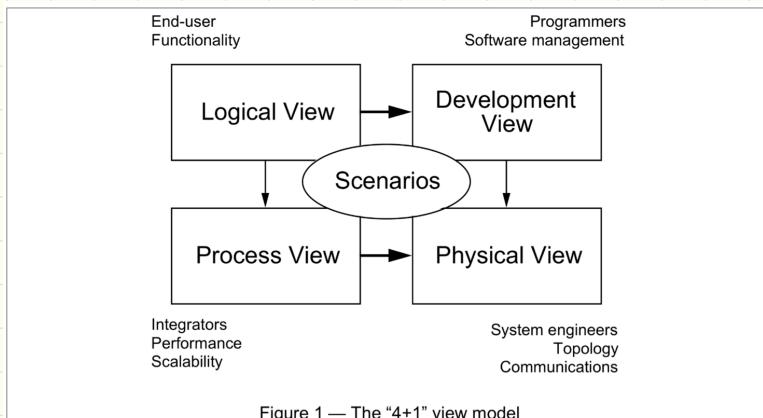


1

CODE IS HARD TO CHANGE

BE VERY, VERY CAREFUL

# Moar DOCUMENTATION !



PLAN  
MORE

2

CODE IS EASY TO CHANGE

ADAPT BETTER

COMMUNICATION

SIMPLICITY

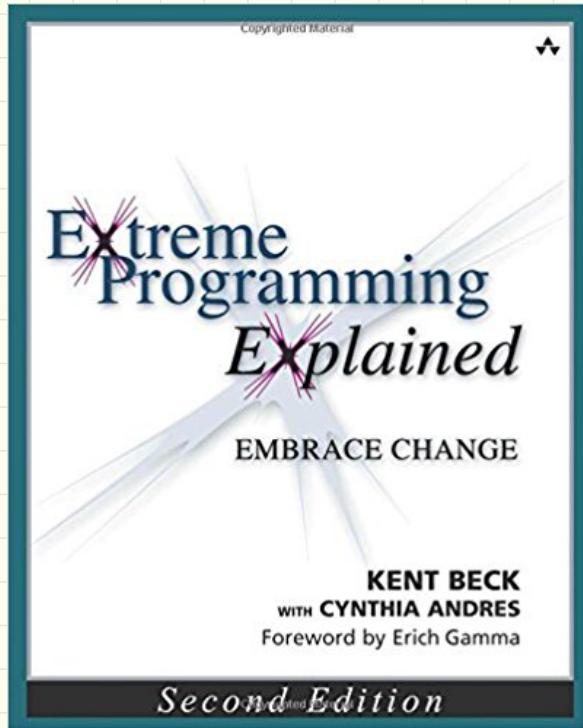
FEED BACK

COURAGE

# History of the Agile Alliance

## “Light” Methodologies:

- Extreme Programming (XP)
- SCRUM
- Feature-Driven Design (FDD)
- Adaptive Software Development
- Crystal
- Pragmatic Programming
- DSDM
- Etc.



- ★ KENT Beck
- ★ Bob MARTIN
- ★ WARD CUNNINGHAM
- ★ Ron JEFFRIES
- ★ MARTIN Fowler

# XP PRACTICES

Customer	Team	Engineering
Define Iteratively	Communicate Iteratively	Build Iteratively

# XP PRACTICES

Customer
Define Iteratively
Short Releases
Planning Game
On-Site Customer

# XP PRACTICES

## ***Team***

*Communicate Iteratively*

Continuous Integration

Sustainable Pace

Metaphor  
(Domain Driven Design)

Collective Ownership

Open WorkSpace

# XP PRACTICES

## ***Engineering***

*Build Iteratively*

Test-First

Simple Design

Continuous Design  
(Refactoring)

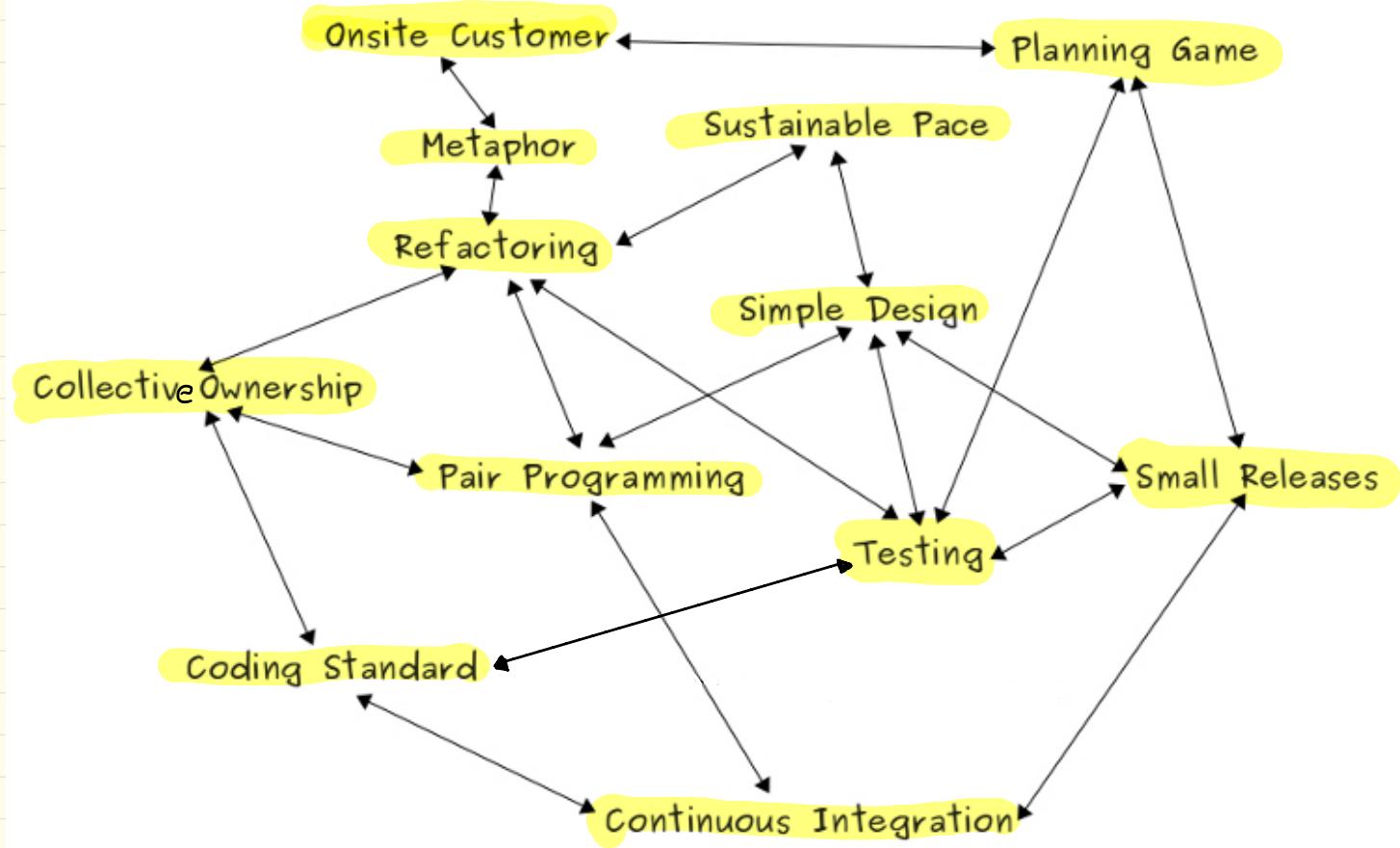
Coding Standard

Pair Programming

# XP PRACTICES

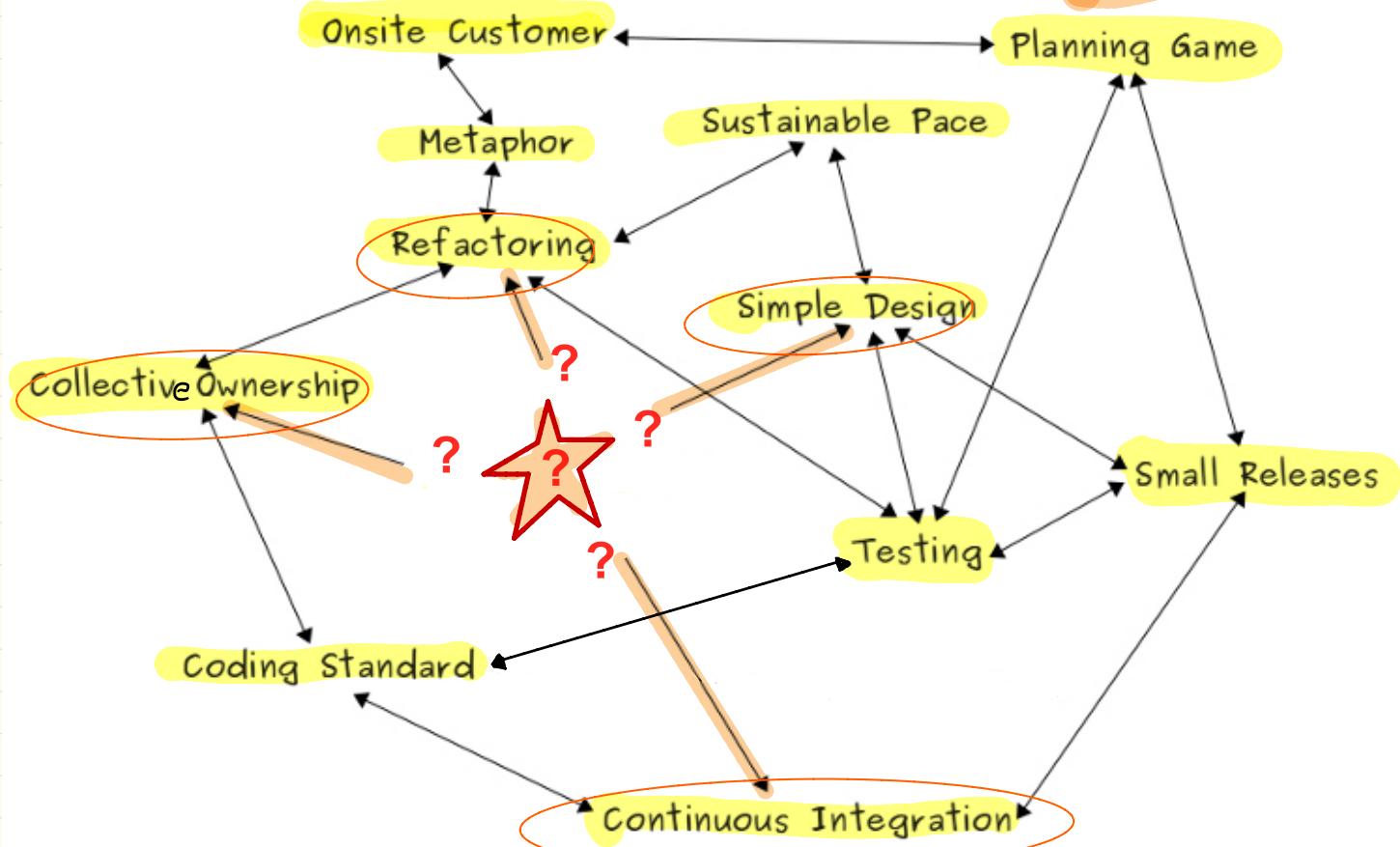
<b><i>Customer</i></b>	<b><i>Team</i></b>	<b><i>Engineering</i></b>
Define Iteratively	<i>Communicate Iteratively</i>	<i>Build Iteratively</i>
Small Releases	Continuous Integration	Test-First
Planning Game	Sustainable Pace	Simple Design
On-Site Customer	Metaphor (Domain Driven Design)	Continuous Design (Refactoring)
	Collective Ownership	Coding Standard
	Open WorkSpace	Pair Programming

# XP web of Practices



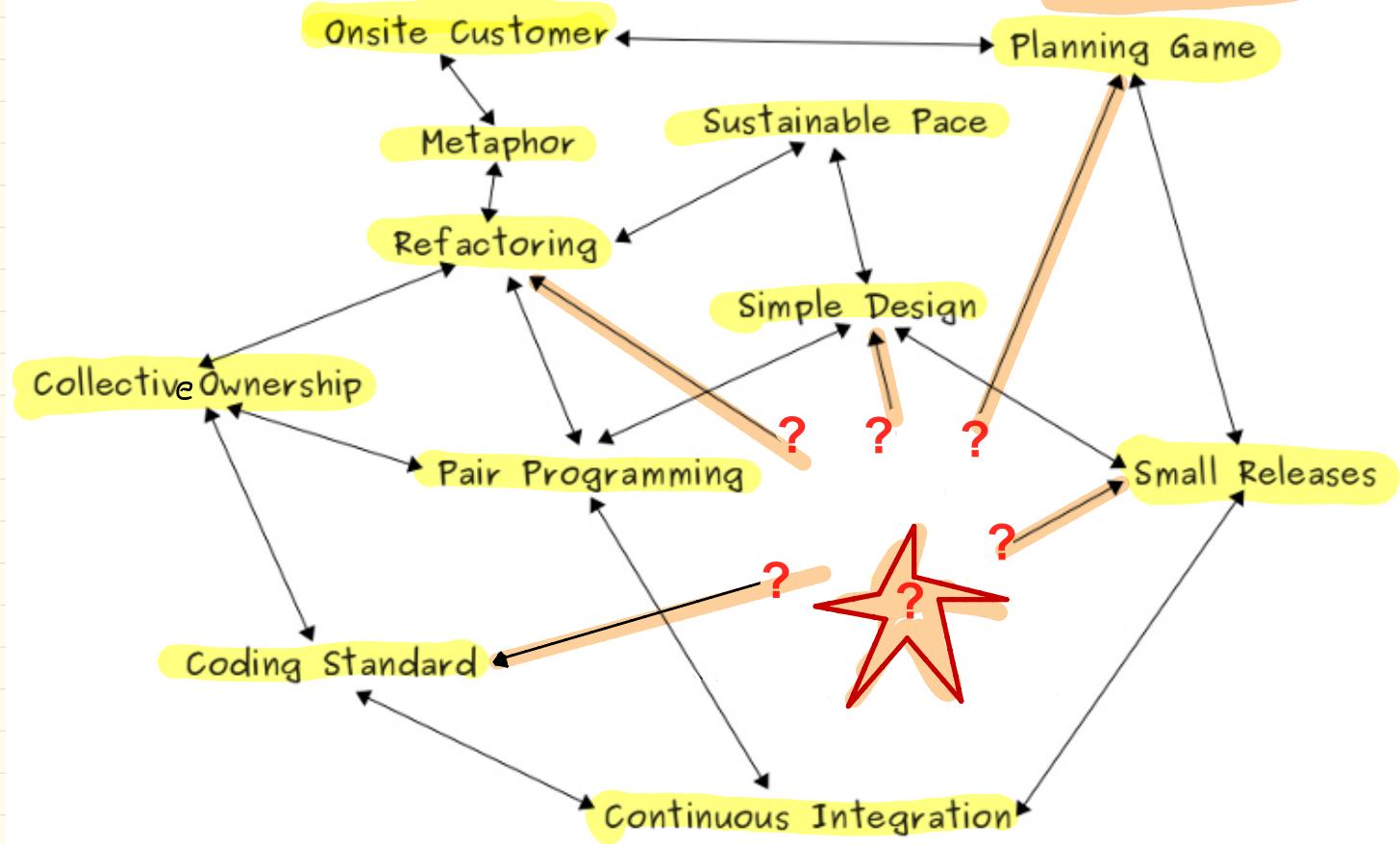
# XP web of Practices

WITHOUT  
PAIR PROGRAMMING

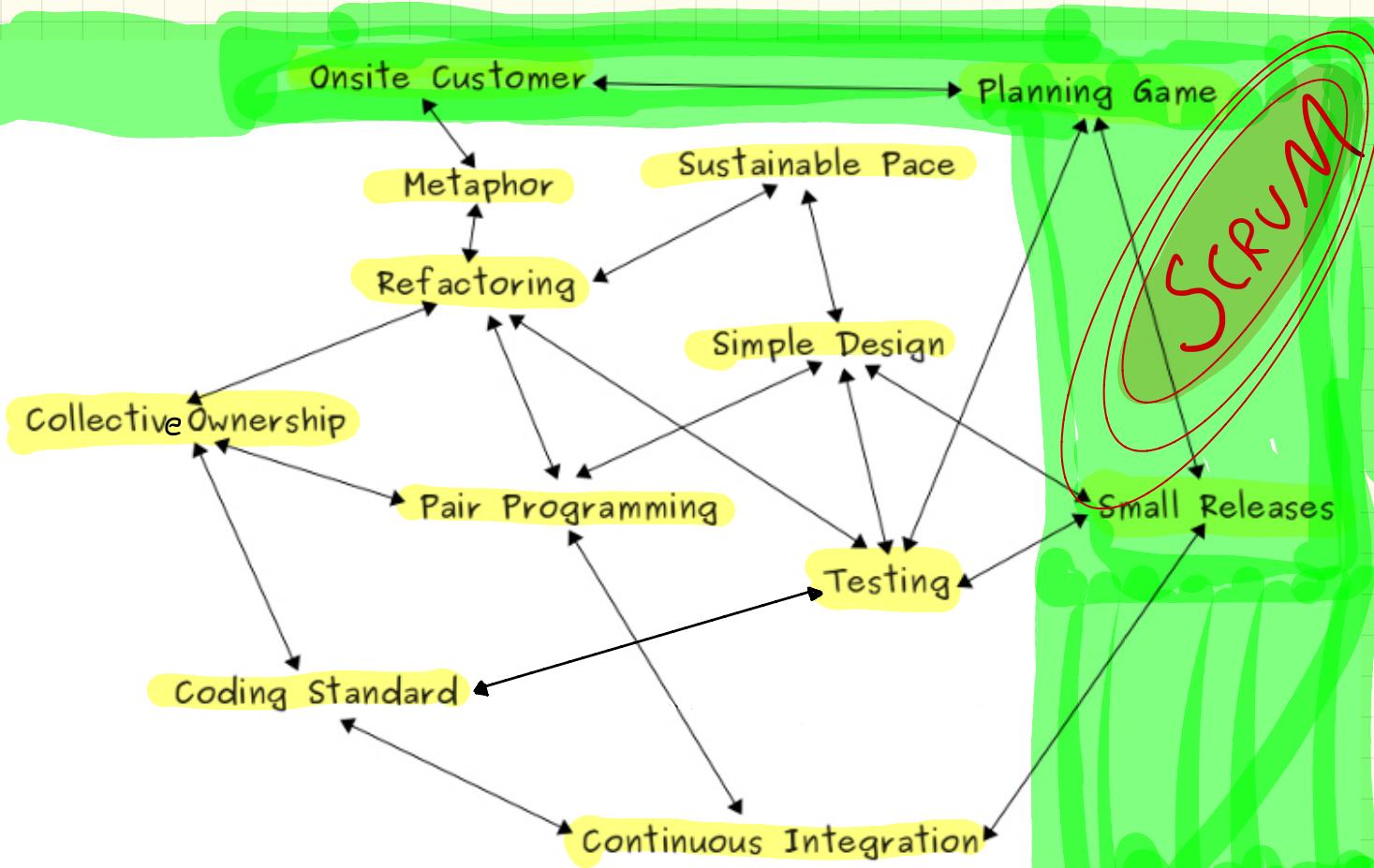


# XP web of Practices

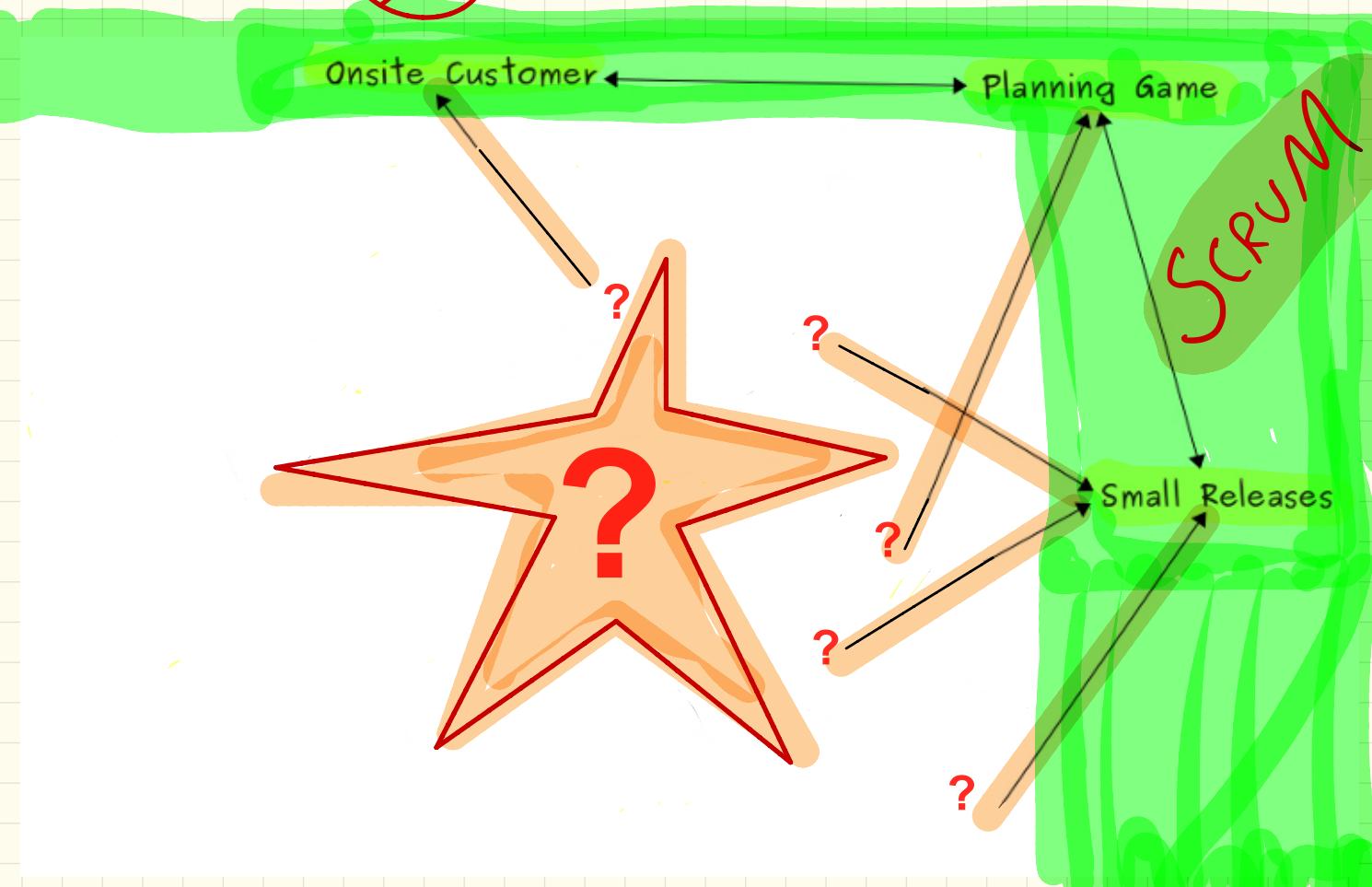
WITHOUT  
TEST-FIRST



# XP web of Practices



# ~~XP~~ web of Practices - SCRUM



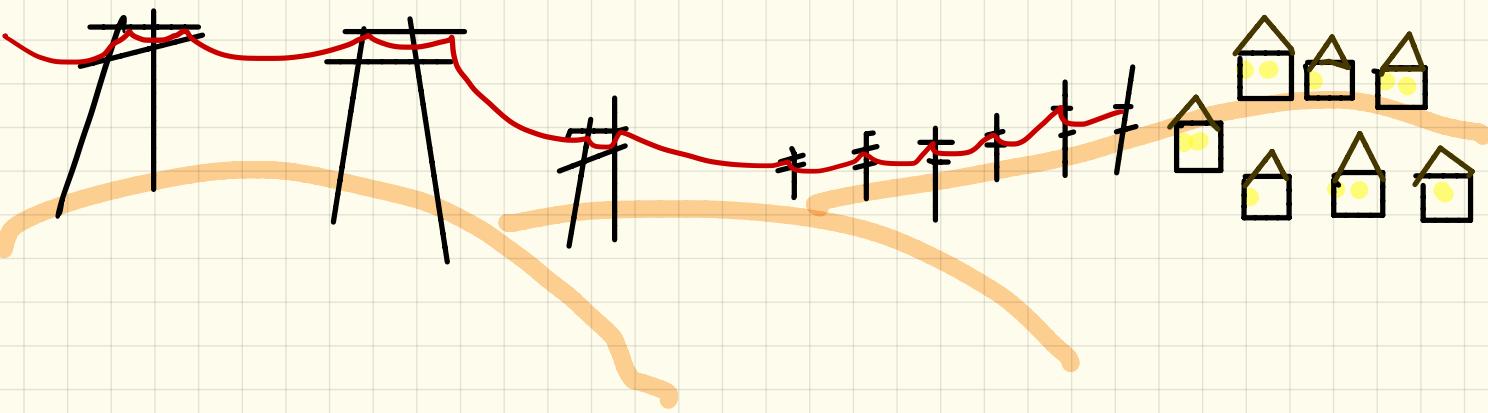
# EXTREME PROGRAMMING



OF BENEFITS  
WHEN FINAL



# THE LAST MILE



# THE LAST MILE

80% OF BENEFITS



SOUNDS

GREAT

WHAT  
HAPPENED

?



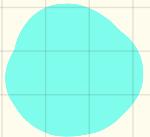
WINDOWS  
XP

BRANDING BLUNDER



BETTER TOOLS

GOOD ENOUGH CODE



CULTURE  
RESISTS



HEGELIAN  
DIALOGIC

WATERFALL

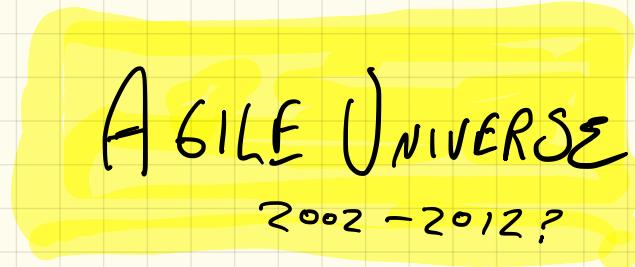
XP

AGILE

# Raise and Fall of eXtreme Programming

---

2001



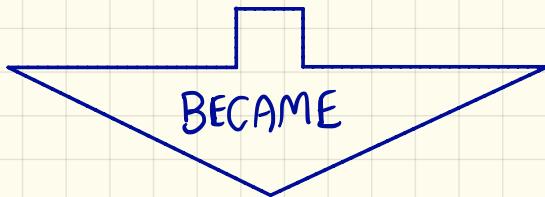
Bob Martin

KEN SCHWABER

XP  
UNIVERSE

PLUS

A GILE  
UNIVERSE



AGILE UNIVERSE

huh?

# The Scrum-ification of Agile

---

Focus ON BUSINESS

---

LOSS of TECHNICAL

---

Bob MARTIN:

What I did not foresee was the loss of the technical emphasis that rapidly followed. The Agile conference has become a conference for project managers, not for programmers

# RETROSPECTIVE

---

WHAT WORKS ?

WHAT DOESN'T ?

# XP PRACTICES

<b><i>Customer</i></b>	<b><i>Team</i></b>	<b><i>Engineering</i></b>
Define Iteratively	<i>Communicate Iteratively</i>	<i>Build Iteratively</i>
Small Releases	Continuous Integration	Test-First
Planning Game	Sustainable Pace	Simple Design
On-Site Customer	Metaphor (Domain Driven Design)	Continuous Design (Refactoring)
	Collective Ownership	Coding Standard
	Open WorkSpace	Pair Programming

# XP PRACTICES

<b><i>Customer</i></b>	<b><i>Team</i></b>	<b><i>Engineering</i></b>
Define Iteratively	<i>Communicate Iteratively</i>	<i>Build Iteratively</i>
Small Releases	Continuous Integration	Test-First
Planning Game	Sustainable Pace	Simple Design
On-Site Customer	Metaphor (Domain Driven Design)	Continuous Design (Refactoring)
	Collective Ownership	Coding Standard
	Open WorkSpace	Pair Programming

# XP PRACTICES

<b><i>Customer</i></b>	<b><i>Team</i></b>	<b><i>Engineering</i></b>
Define Iteratively	<i>Communicate Iteratively</i>	<i>Build Iteratively</i>
Small Releases	Continuous Integration	Test-First
Planning Game	Sustainable Pace	Simple Design
On-Site Customer	Metaphor (Domain Driven Design)	Continuous Design (Refactoring)
	Collective Ownership	Coding Standard
	Open WorkSpace	Pair Programming

TEST FIRST

SIMPLE DESIGN

REFACTORING

DOMAIN DRIVEN DESIGN

INTERNAL

QUALITY

I.Q.

# Quality

EXTERNAL → EFFECTS USER

INTERNAL → EFFECTS CODER

↳ SPEED & QUALITY

# Correlation Premise

— MIKE HILL

REDUCED  
QUALITY

EXTERNAL →

DELIVER

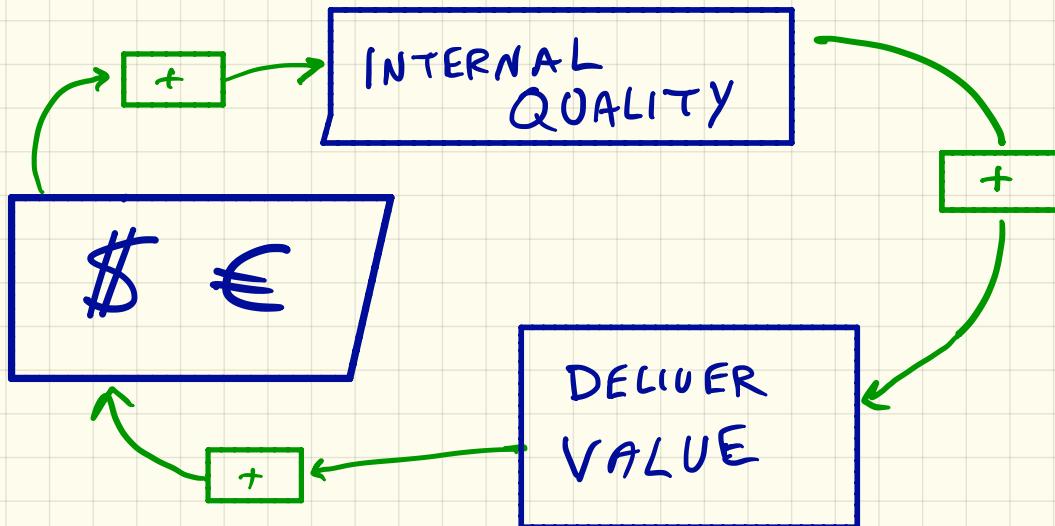
FASTER

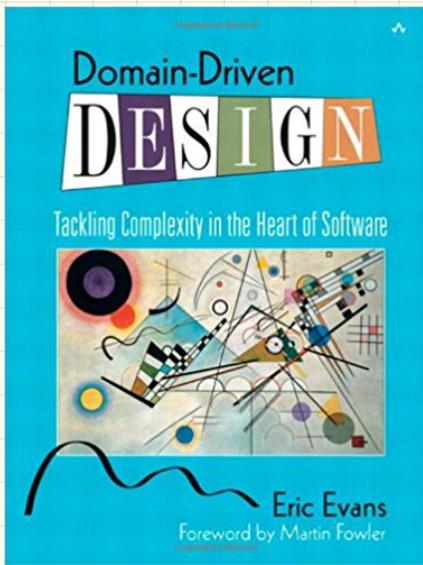
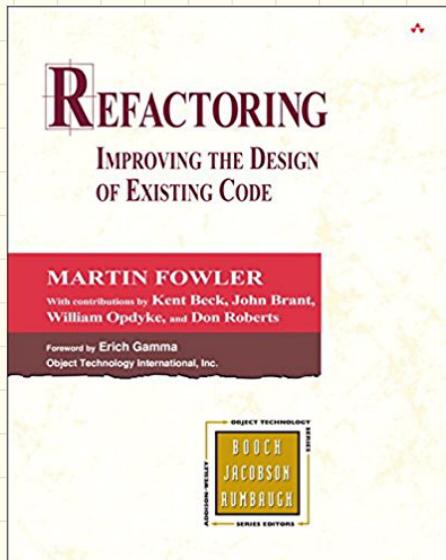
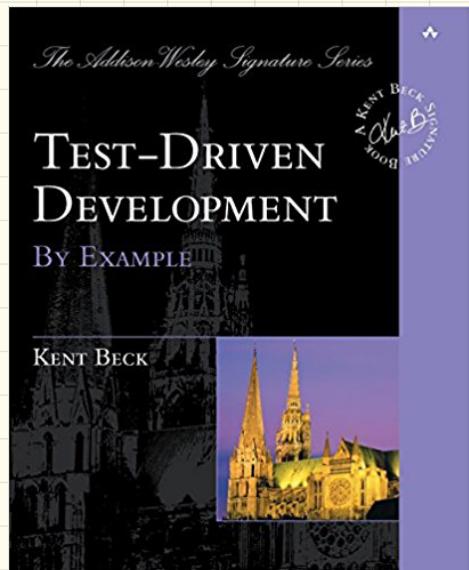
INTERNAL →

SLOWER

# THE LAST AGILE MILE

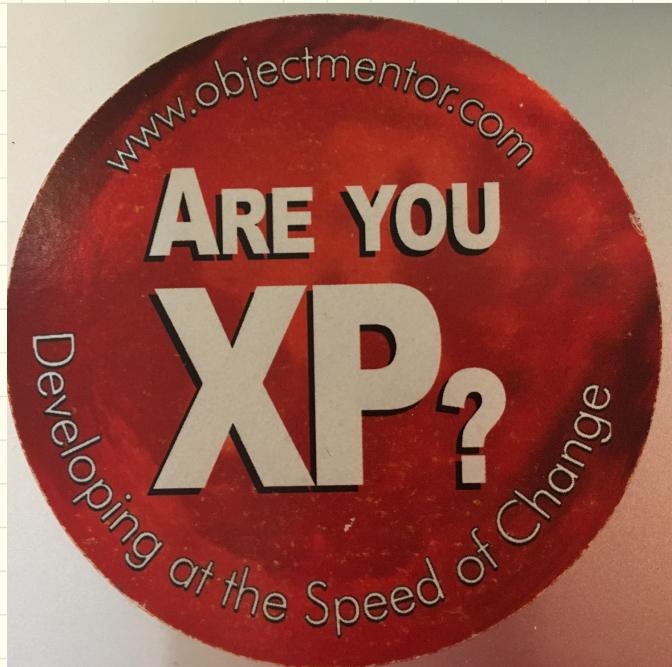
80% of Benefits with Last 20% of Practices





# Peak Productivity

CIRCA  
**2000**



## AGILE MANIFESTO

**Individuals and interactions**

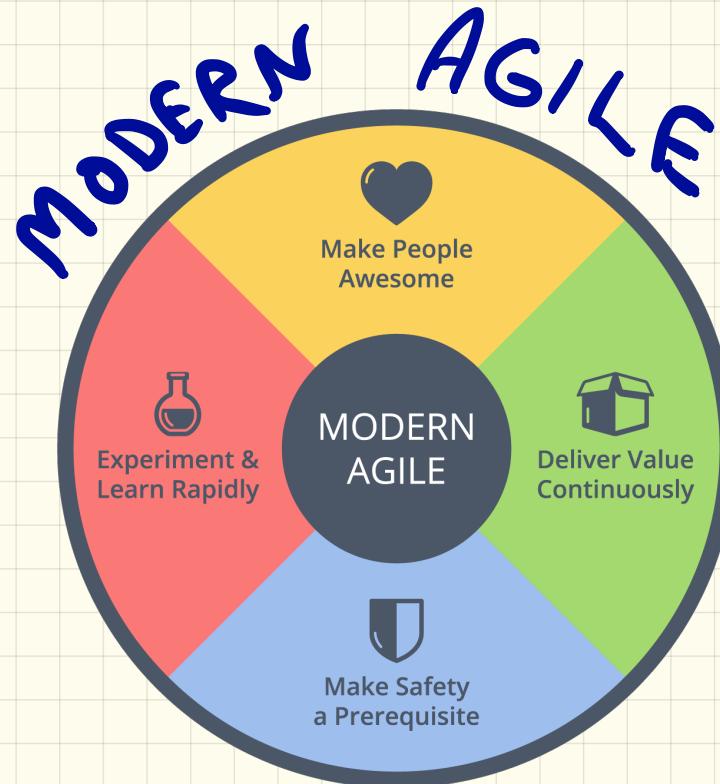
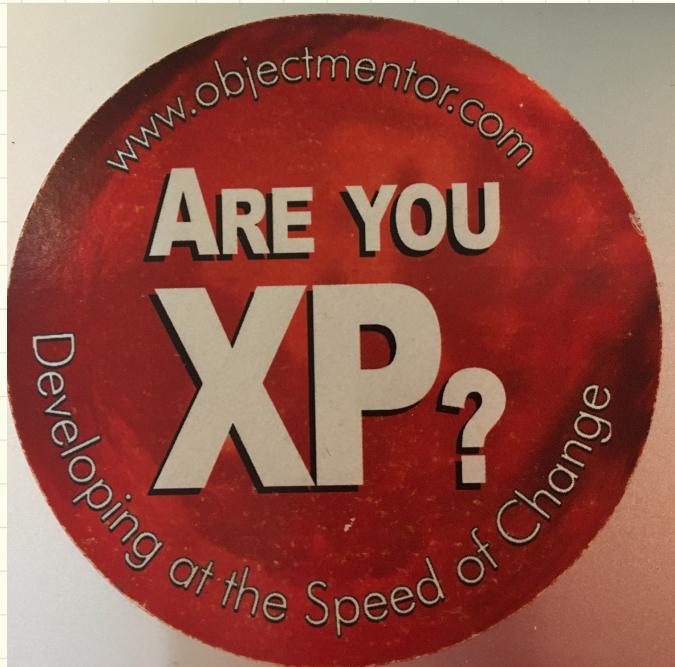
**Working software**

**Customer collaboration**

**Responding to change**

# Peak Productivity

CIRCA  
2000



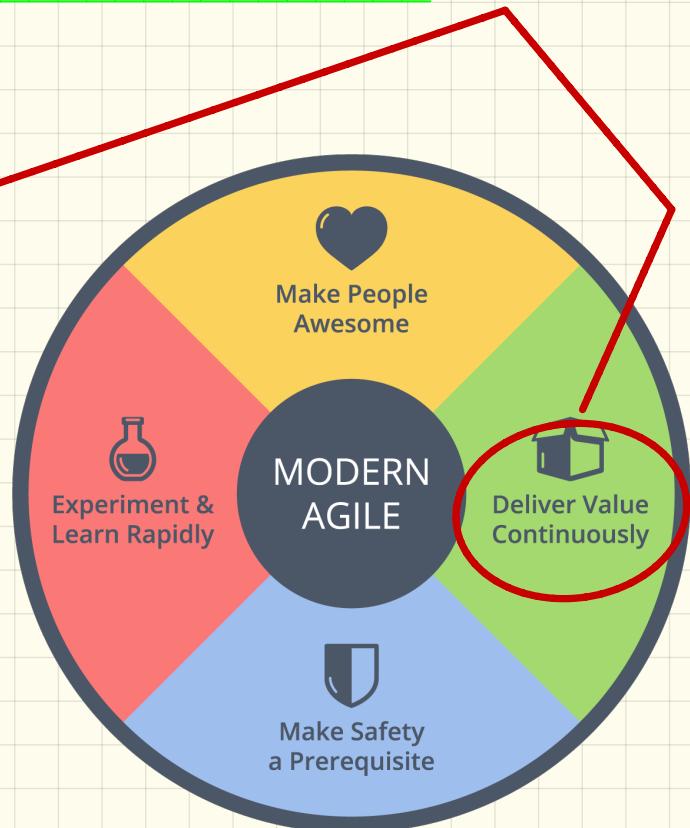
# Agile and Modern Agile

Working software

Customer collaboration

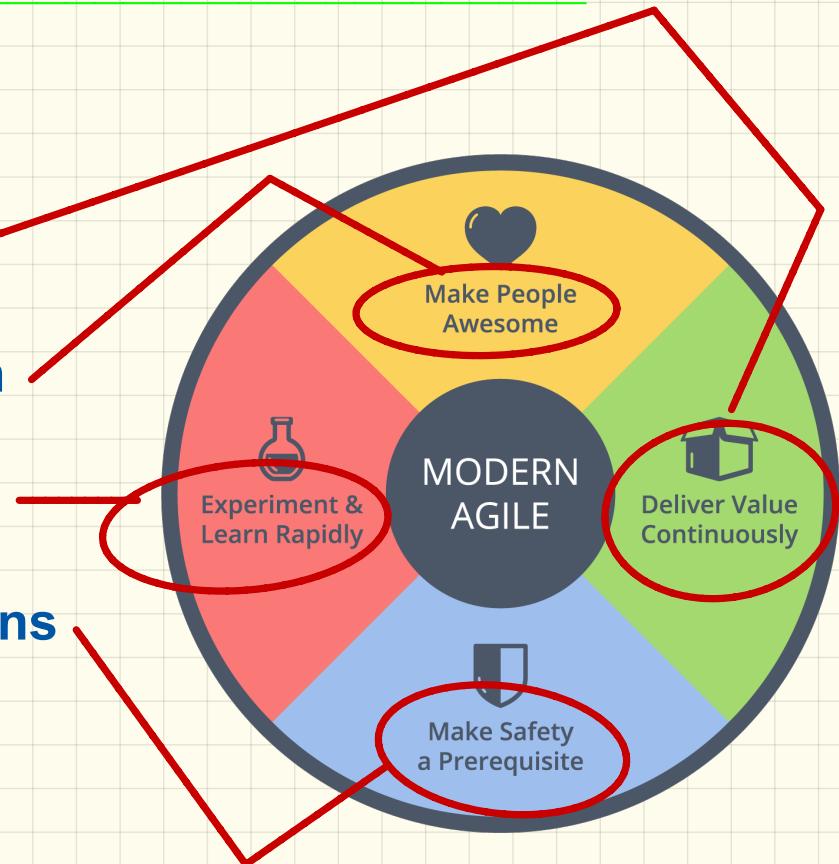
Responding to change

Individuals and interactions

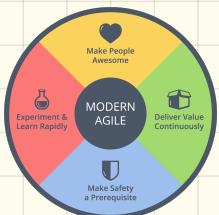


# Agile and Modern Agile

- Working software
- Customer collaboration
- Responding to change
- Individuals and interactions



2000  
Now



SCOPE

TEAM

COMPANY

DELIVERY

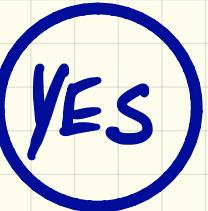
ITERATIVE

CONTINUOUS

# **Extreme Programming**

## **Resurgent? Or Retro ?**

# Extreme Programming Resurgent? Or Retro ?



# Extreme Programming Resurgent? Or Retro ?

YES

RESURGENT →

LAST MILE PRODUCTIVITY

HOWEVER

RETRO →

STILL JUST OLD AGILE

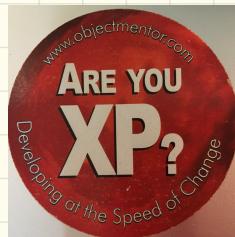
# Extreme Programming Resurgent? Or Retro ?



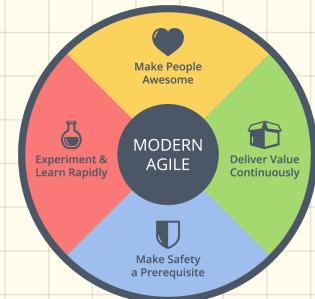
RESURGENT

IMPROVE I.Q.  
DELIVER MORE

RETRO



ON TO



# That's all I've got

[github.com/mwindholtz/presentations](https://github.com/mwindholtz/presentations)

