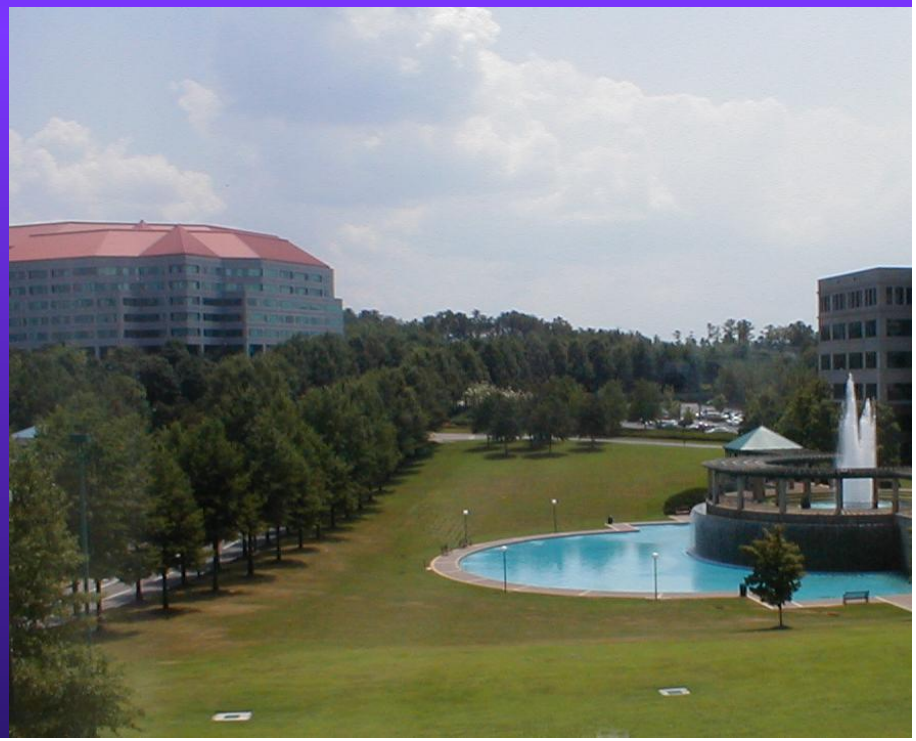
A Day In The Life
of an Extreme Programmer

Mark Windholtz
ObjectWind
Software Ltd



Who Am I

- Mark Windholtz
- ObjectWind Software Ltd
- 14 years Building Professional Software
- Business, Medical, Engineering, Military
- First contact with XP in Oct 1996
- Started Six XP teams
- Consulted for Eight other XP teams

Agenda -1

1. **Why XP**
2. *A Day in the Life*
3. *Why does it work?*

Standard Industry Practice

- Most Software development is chaotic
- "Code and Fix"
- Indicators
  - Long Integration Phase
  - Long Test Phase
  - High Defect Counts
  - "Second Attempt" Analysis Paralysis

Solution of the 1990's

- Rigorous Heavy Process
- Lots of Documents
- Lots of Control Points
- Blue Prints before you build

Why not common practice?

- Not Very Successful
  - *if only our people would follow this great process!*
- Conflicting Goals
  - Process vs. Product
- Criticized as bureaucratic & complicated
  - *What's harder: building the s/w or implementing RUP?*
- Big Processes often introduce added organizational conflict

Extreme in the Middle

- Compromise
  - Between No Control and No Movement
  - Between Chaos and Over Complicated

Agenda -2

1. Why XP
2. **A Day in the Life**
3. *Why does it work?*

Delivering to Deadline

XP is about delivering to deadline.
Every two weeks you release something
and every two weeks you check your plan.
You get really good at getting stuff done.

- James Grenning, ObjectMentor

Team Planning

Iteration Planning

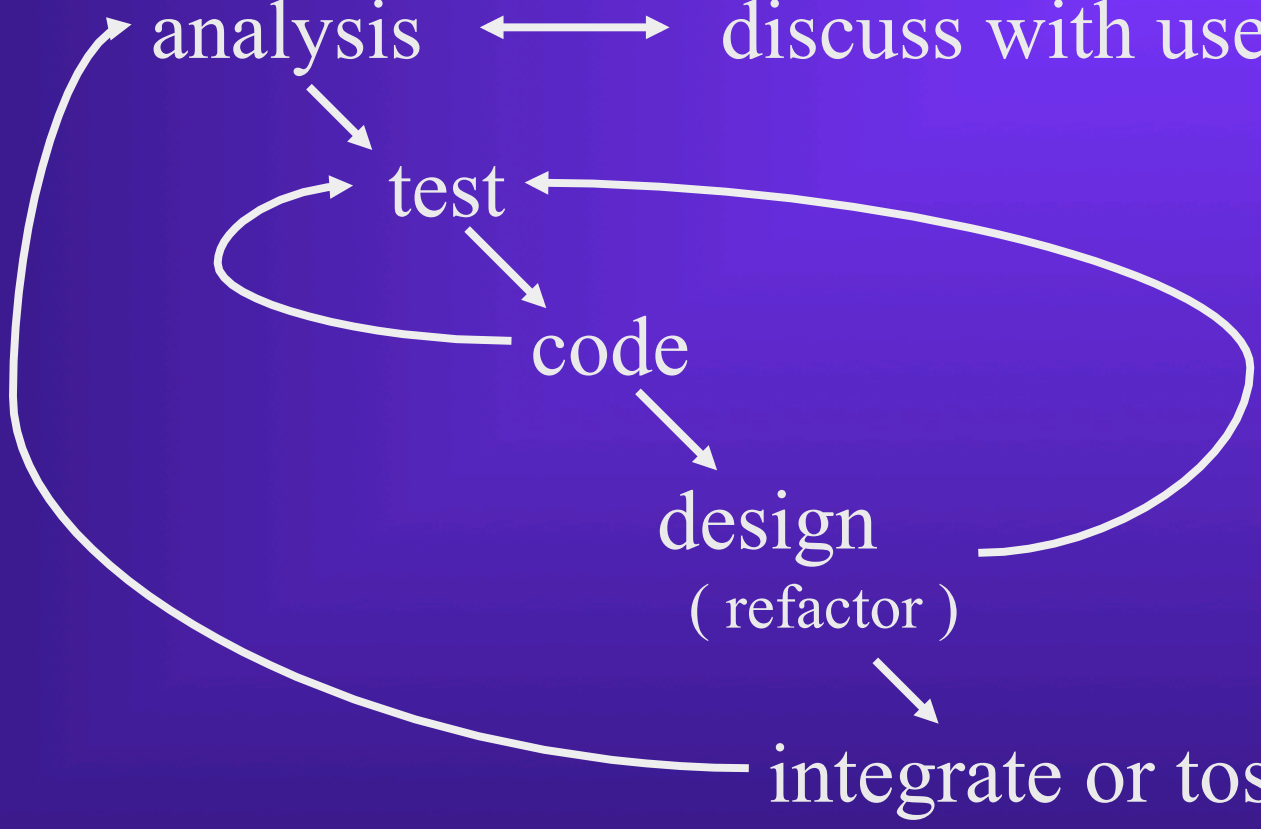
A Day in the life of XP

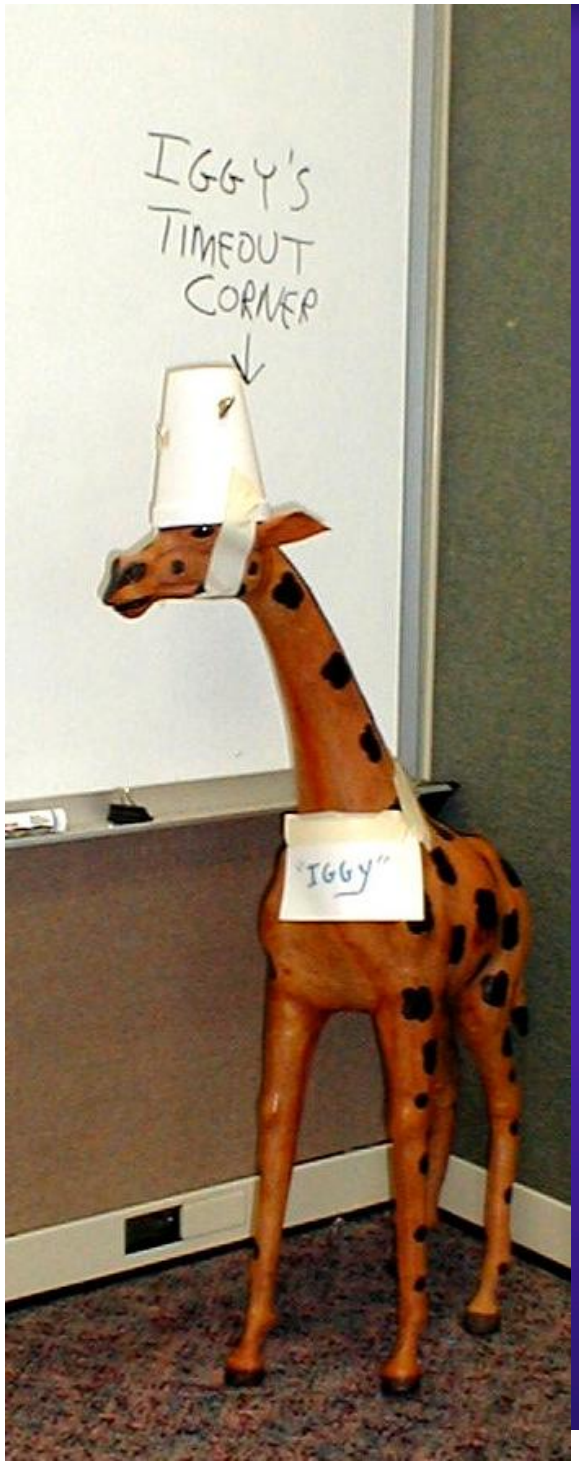
- 15 minute team stand-up meeting.
- Pair-up on a task.
- Go to the open workspace
- Where the customer sits with the team
- A number of Programming episodes.
- Improve any code that smells bad.
- Go home at a reasonable time.

The Task Board

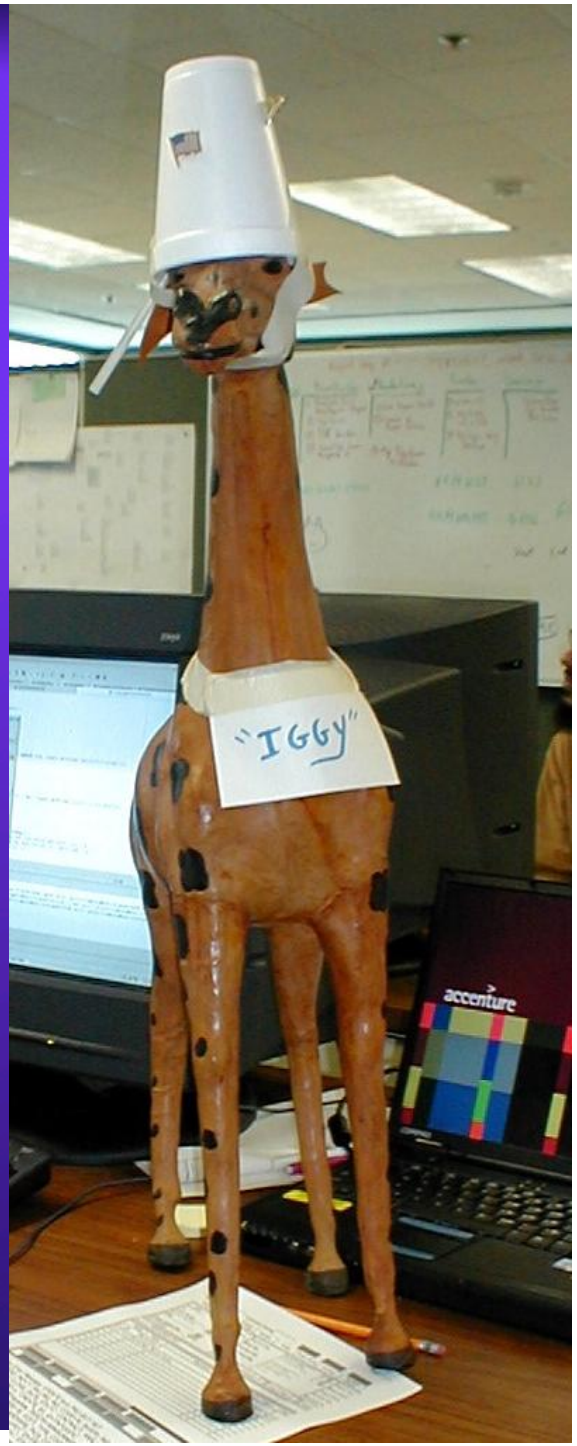
Open Workspace

XP Episode Cycle





Integration "Token"



Core Practices

Customer

How to
Define Features Iteratively

Small Releases

Planning Game

On-site Customer

Programmer

How to
Build Features Iteratively

Test-First

Pair Programming

Continuous Design Improvement

Simple Design

Coding Standard

Team .

How to
Communicate & Schedule

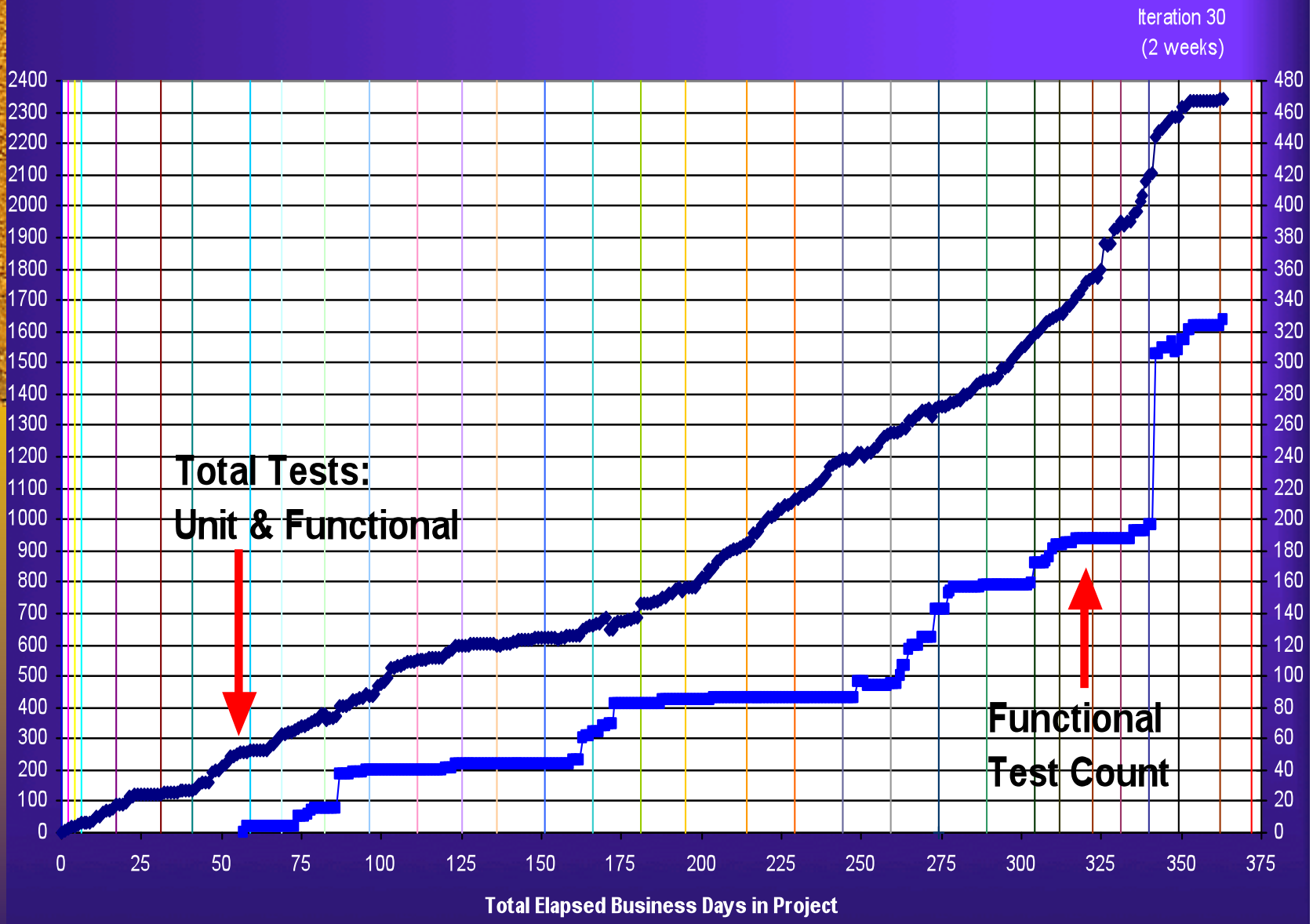
Open Workspace

Continuous Integration

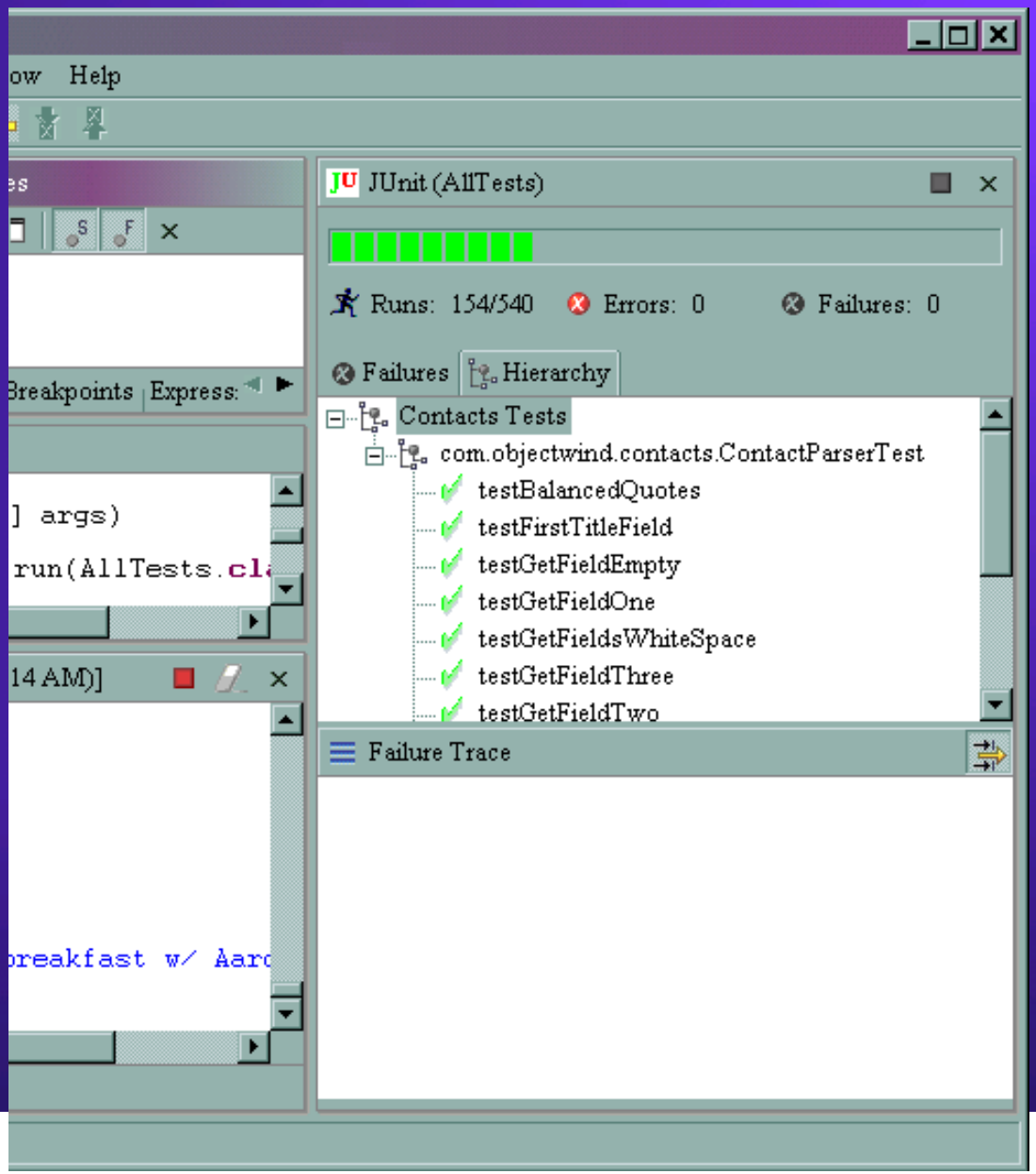
Collective Ownership

Sustainable Pace

Unit & Functional Tests



JUnit TestRunner in Eclipse



Pair Programming

- Code review is good,
- So review all code as it is typed in
- One person types (tactical)
- One person thinks (strategic)

Pair Programming Photos

Pair Programming Photos

Agenda -3

1. Why XP
2. *A Day in the Life*
3. **Why does it work?**

Simple Process

- People talking together, and learning.
- Tested and running clean code.
- Short term measured goals.
- Practice of Agility in everything.
- As Japan taught Detroit
  - Quality can be free

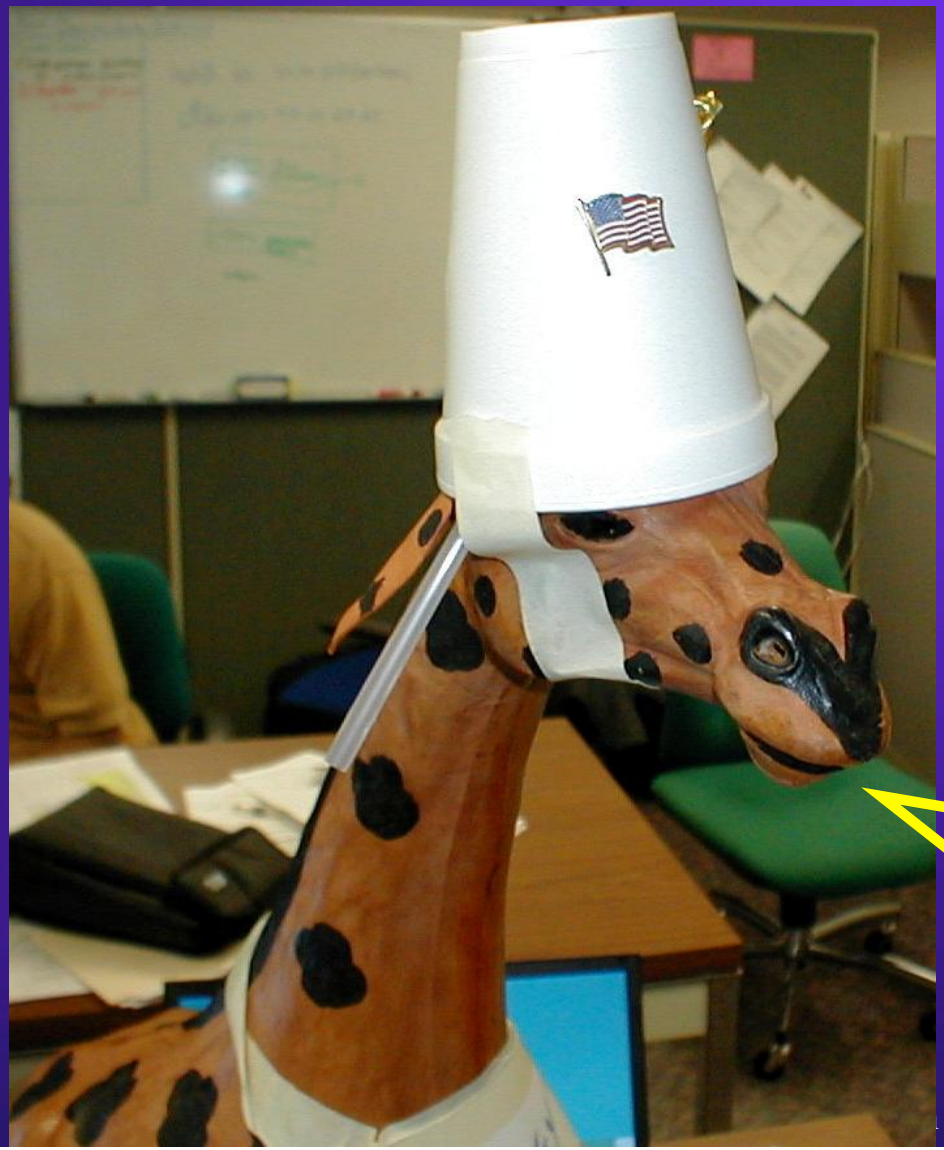
Quality code faster to produced

- Quality
  - Pair Programming
  - Unit testing, Acceptance Testing
  - Anyone can clean the code
- Less Backtracking
  - 80% Less debugging
  - Dramatically Less Defects
- Clean, Tested code is easier to change than a UML model.

Summary

- Why
  - There was chaos
  - Big design produced a poor record
- What
  - A simple, more predictable way to work
- Why does XP work?
  - Supports Complex Systems approach
  - Simple Process
  - Quality code is faster to build

Say “Goodbye” Iggy



OK. Goodbye, Iggy.