

# Herding Cats

**Managing the Software Development Process at Microsoft**

James Helfrich – Professor of Computer Science at  
Brigham Young University – Idaho

# Background

## Education:

- Bachelors degree in Computer Science in Engineering at the University of Pennsylvania
- Masters degree in Computer Information Systems at the University of Pennsylvania
- (in progress) PhD in Instructional Design at Idaho State University

## Vocational Experience:

- Lieutenant (retired) of the United States Navy
- Developer on Microsoft Word ('95, '97, 2000, XP, 2003, 2007)
- Program Manager on Microsoft Publisher (2007)
- Professor of Computer Science (Software development, Computer Security, Human-Computer Interaction)



# ROADMAP

What we will talk about today

- The Context
- A Historical Perspective
- The Big Questions
- Q & A

# THE CONTEXT



The Software Industry  
The Software Development Process  
The Developer (aka: the cats)

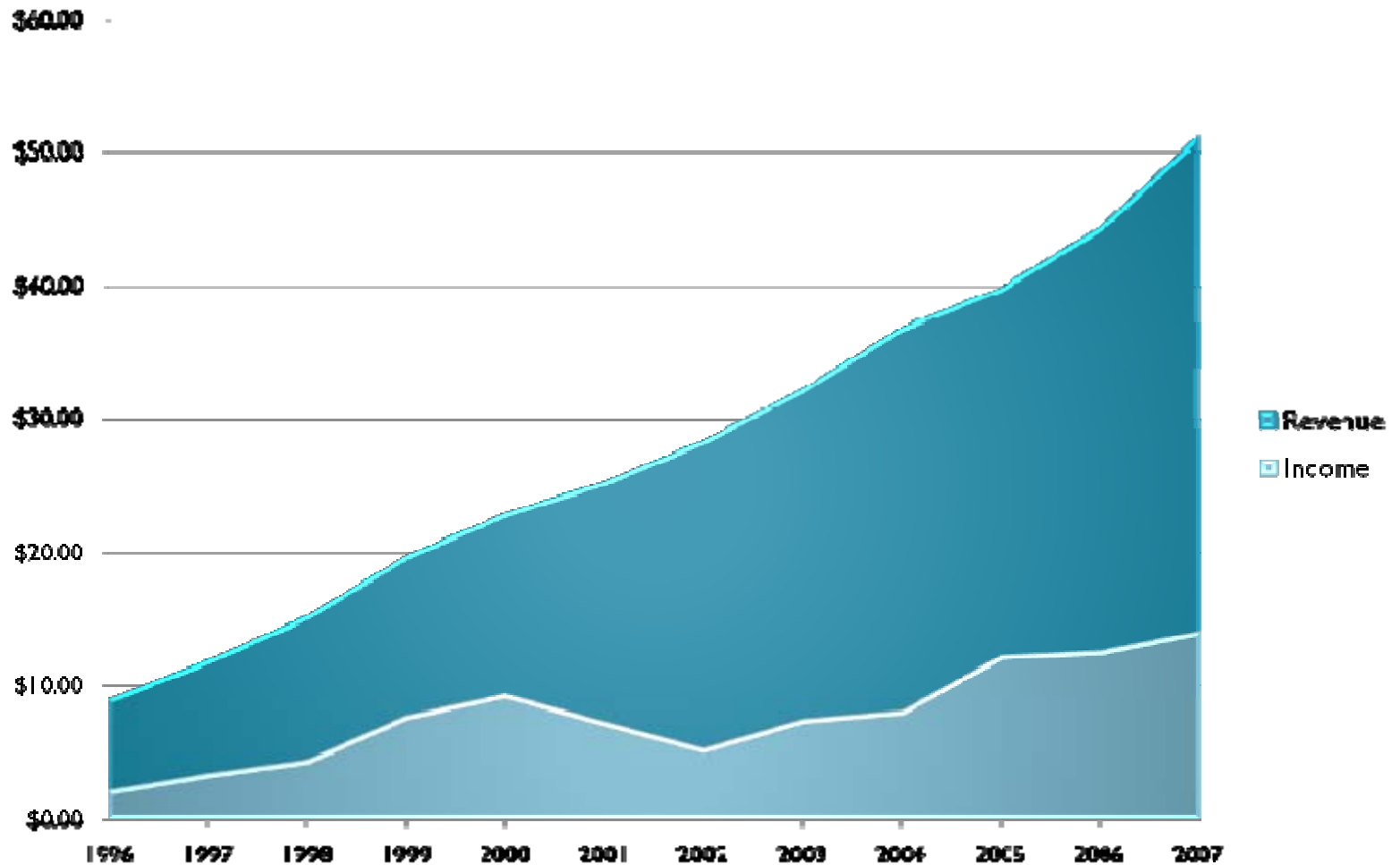


# **SOFTWARE INDUSTRY**

There are only two things that are unique about the software industry:

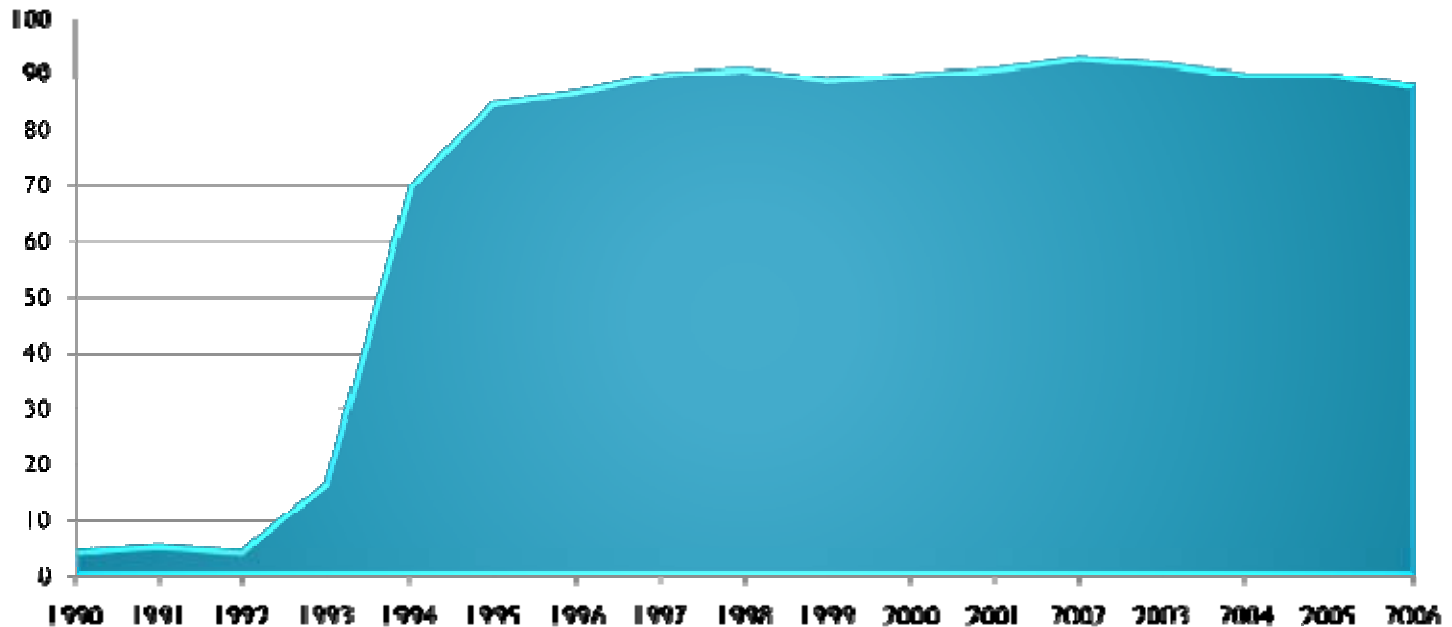
1. The pace of change is very fast
2. The field is extremely immature

# Size of the software business





# Tendency for "Standardization"



Market share by year for Microsoft Word from 1990-2006.

A similar graph exists for:

**Google**

**PowerPoint**

**Excel**

**Internet Explorer**

**Outlook**

**Word Perfect**

**Photoshop**



# PROJECT

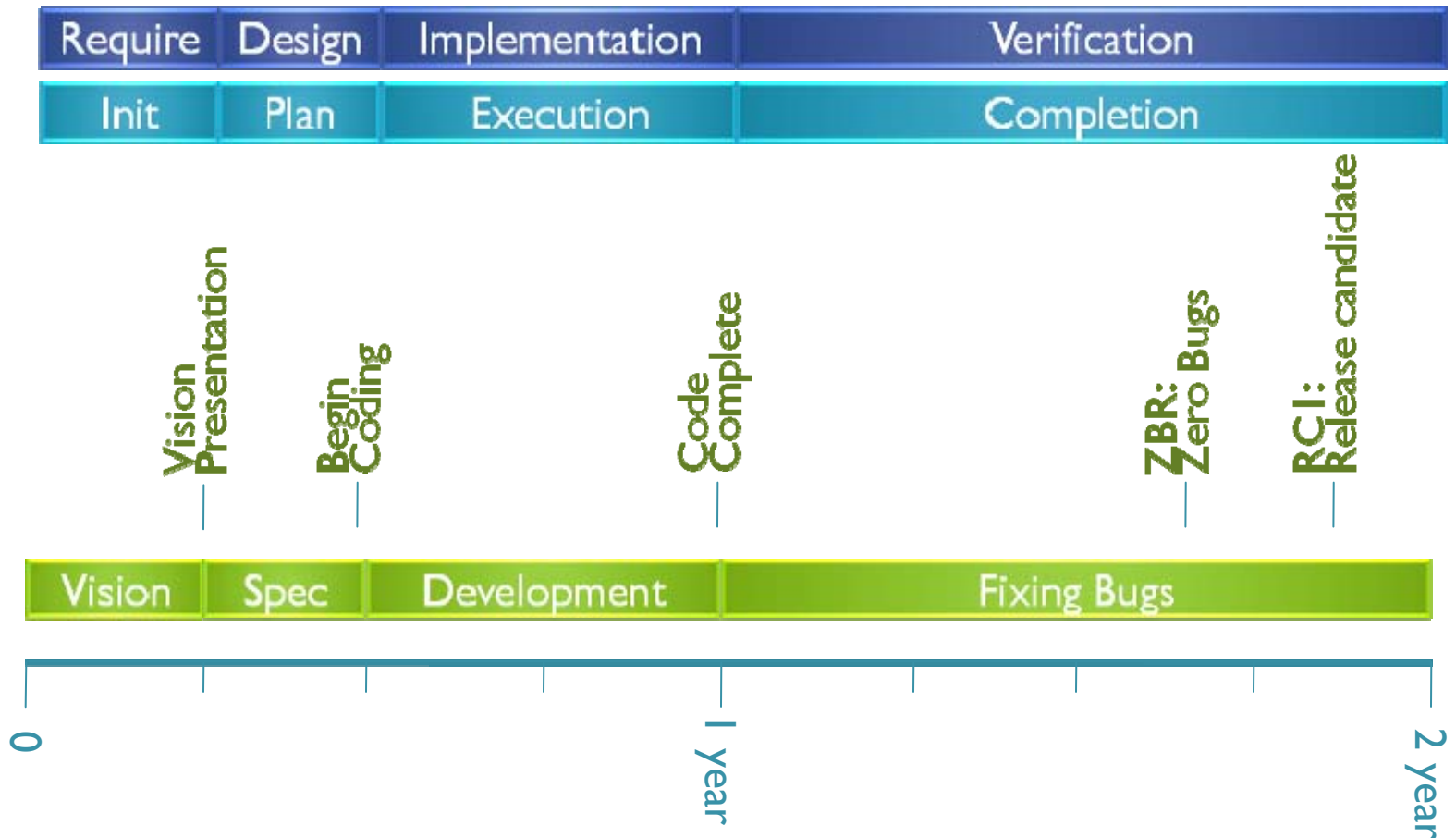
It turns out, defining a project in the software engineering space is not trivial. Everything is a project, but nothing really is.



# A "Project"

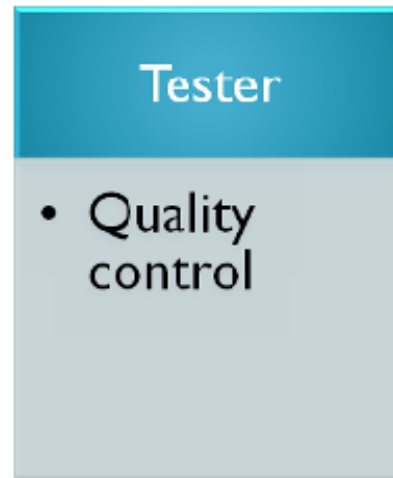
- So, what is a project?
  - A single product release?
  - A single feature team?
  - A single unit of work?

# Project: One Release



# Project: One Feature Crew

- Three main disciplines



- Purpose: complete one feature
- Duration: possibly 3 days, possibly 3 years

# Project: Single check-in

- What?
  - Every change is made by a developer as a check-in
  - This constitutes adding, removing, or changing the source-code
- Why?
  - Implement part of a feature
  - Fix a bug
- When?
  - With every bug fix
  - With every milestone in feature work
  - Periodically: weekly or monthly





# DEVELOPERS

One of the most critical things about understanding the software engineering space is to understand the developer.


# Pre-Madonna

- Traditionally, developers are the most influential member of the feature crew
- Every developer has their own techniques and style
- Developers are highly independent
- So, what is it like working with developers?

# Common Developer Fallacies

- Management
  - Managers tend to get chosen from developers, though they tend to have little social skills and less appreciation for the problem
- Cowboy philosophy
  - "We can fix that; it's just software"
  - "Give me a weekend and a box of pizza and I will whip out a solution for you"
- Pride
  - "You can't rip that code out; it works fine!"

# A HISTORICAL PERSPECTIVE



Pyramid & Bug Counts  
T3 & Workaholic Wednesdays  
'95 & Pearl Harbor  
'97 & Reconcile  
2007 & Morpheus





# **Problem: Measurement**

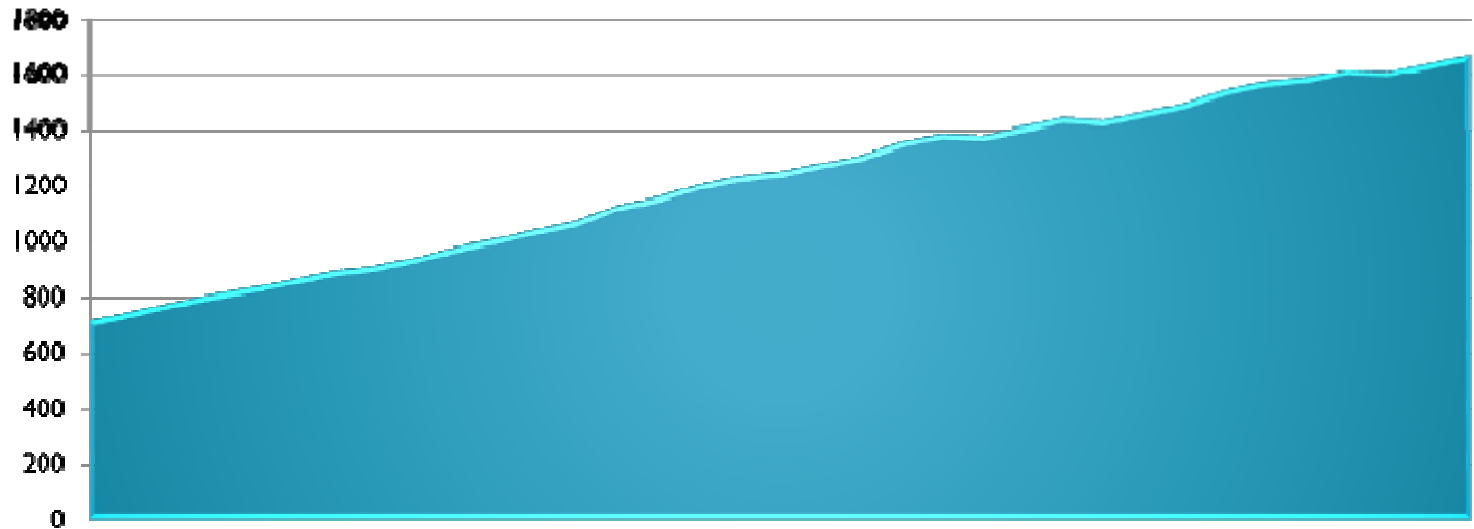
How do you measure the status of a software project

# Pyramid: context

- Historic context
  - After the successful release of Word 2.0 (Spiff)
  - Want to re-architect the product

# Pyramid: problem

- Bug Count
  - Each week, the bug count grew by about 30
  - At what point would we hit zero bugs?



# If all's you got is a hammer...





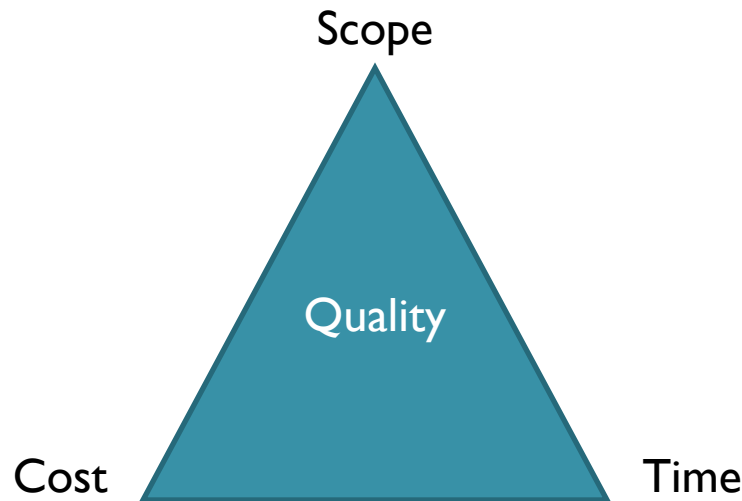


# **Problem: Setbacks**

How do manage constraints?

# T3: Work-aholic Wednesdays

- Problem
  - The bug list is "out of control" aka Pyramid
- Solution
  - "Everyone works until their bug goals are met"



# 97: Reconcile

- Background
  - One of the big 5 features of Word '97
  - One "senior" developer was solely assigned
- Lead-up
  - 1/4 of the bugs were for the Reconcile feature
  - Each bug was very old, and very severe
- Fallout
  - Eventually 1/3 of the team was assigned to the area
  - Eventually, the entire feature got cut
- Why?



# **Problem: Adaptability**

How do you turn an ocean liner on a dime?



# 95: Pearl Harbor

- Problem:
  - Microsoft is a huge company with 20,000 "highly independent" workers
  - The internet is becoming a big deal, and Microsoft is not in a strong position
- Solution
  - Bill Gates gives his infamous "Pearl Harbor" speech
- Result



# **Problem: Communication**

How do you herd all those cats?

# 2007: Morpheus

- What
  - One of biggest features in Publisher 2007
  - It had a cool name
- Problem
  - Unknown goal / purpose
  - Not stabilizing
- Solution
  - ??



# THE BIG QUESTIONS



How can you possibly bring order to this chaos?

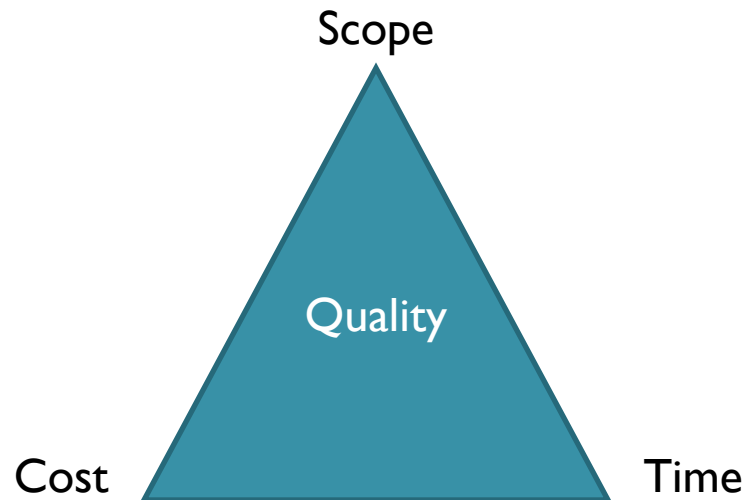
# Question 1: Measurements

- How can you determine the status of a project?



# Question 2: Setbacks

- What do you change when things are going poorly?



# Question 3: Adaptability

- You do you maintain agility that is critical in a rapidly changing environment?

# Question 4: Consensus

- How do you get everyone to agree?



# QUESTIONS & ANSWERS