Introduction to Extreme Programming

Brian Button
bbutton@objectmentor.com
Object Mentor, Inc

Overview

- Motivation
- Business Case
- Description
- Case Studies
- References

My Typical Client

- Manager doesn't trust programmers
  - always late
  - low quality
- Programmers don't trust manager
  - unreasonable expectations
  - always changing their minds
- Cube Farm
- Us versus Them
- Bad code

My Job

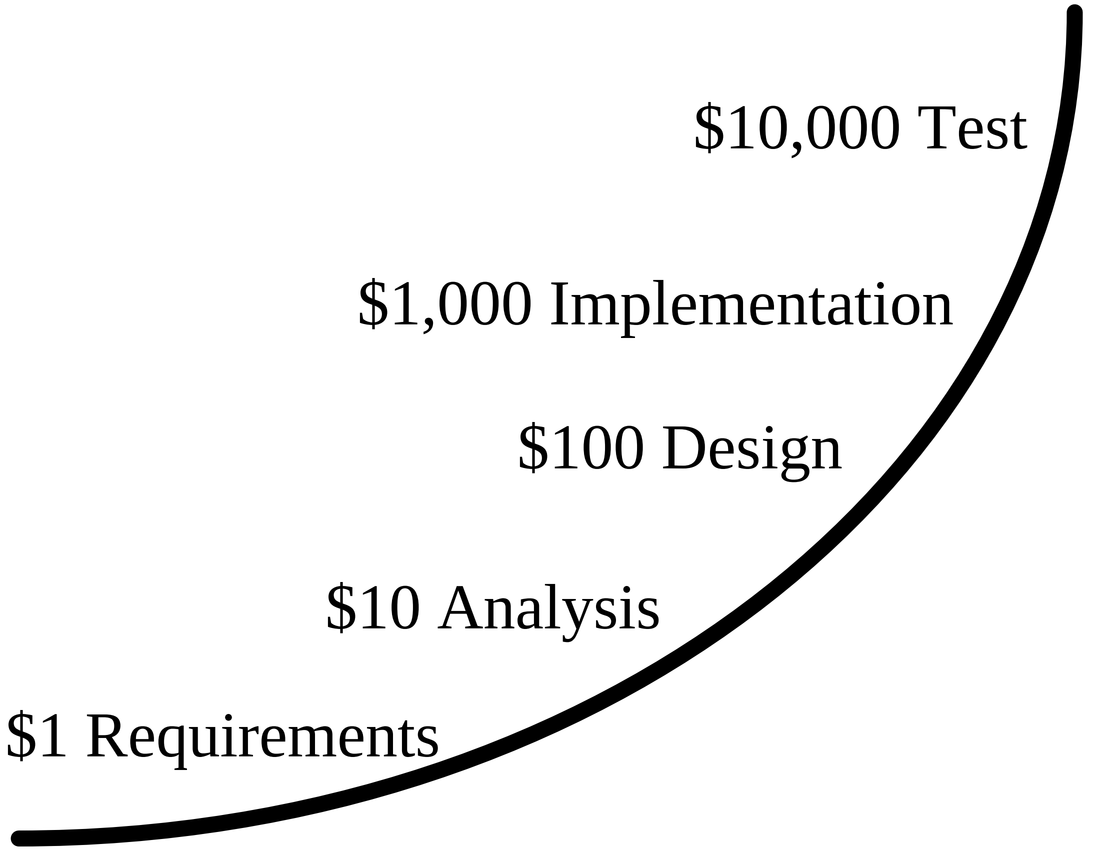
- Trust between management and programmers
  - Quick wins
  - Freedom
- Improve quality
  - Internal
  - External
- One Team
- Align authority with responsibility

My Goal

Communication

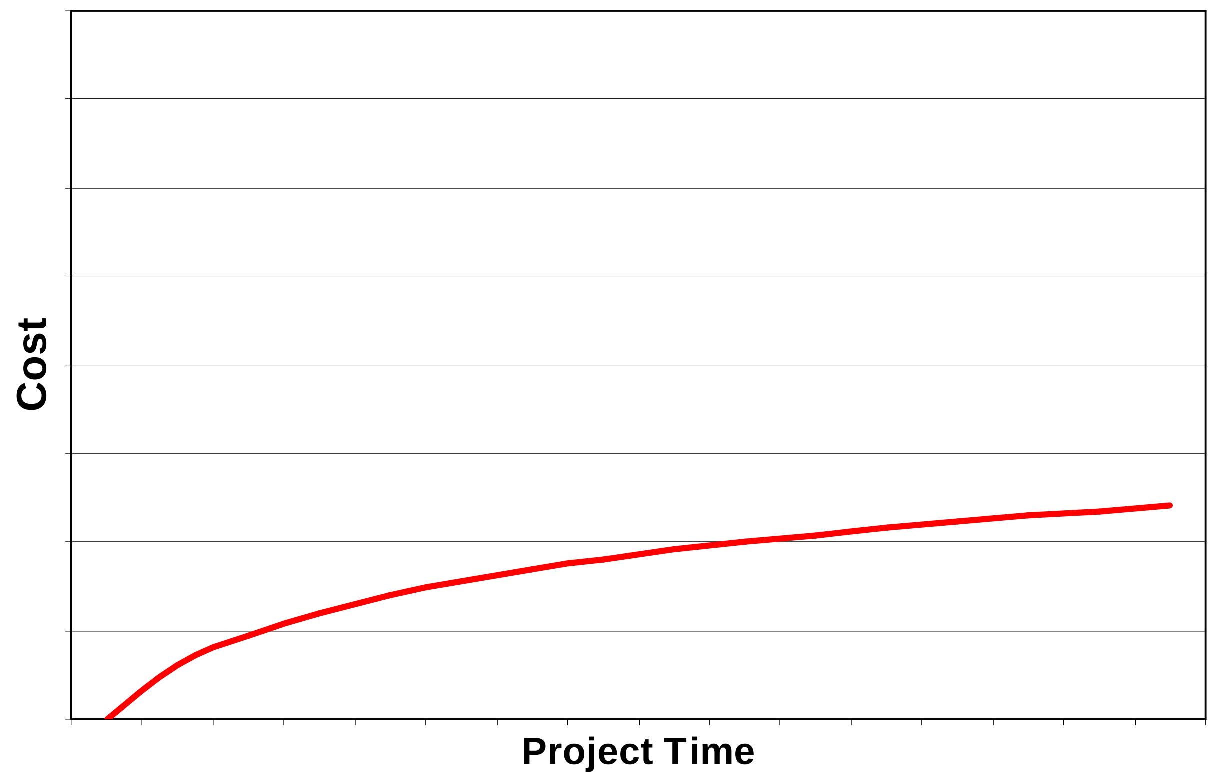
V1.0

Traditional View of Development



Cost of change thought to grow exponentially with time

What if ...



XP works to make this curve flat for most of a project?

Then I could ...

- Analyze and design one feature at a time
- Change my mind
- Delay decisions
- Work on most important feature
- Focus on customer-visible work rather than infrastructure
- Manage risk

Benefits

- Working system early, not designs on paper
- Quantifiable progress against project plan
- Flexibility to add, drop, or change features at (almost) any time
- Defer decisions until risk is reduced

What is XP?

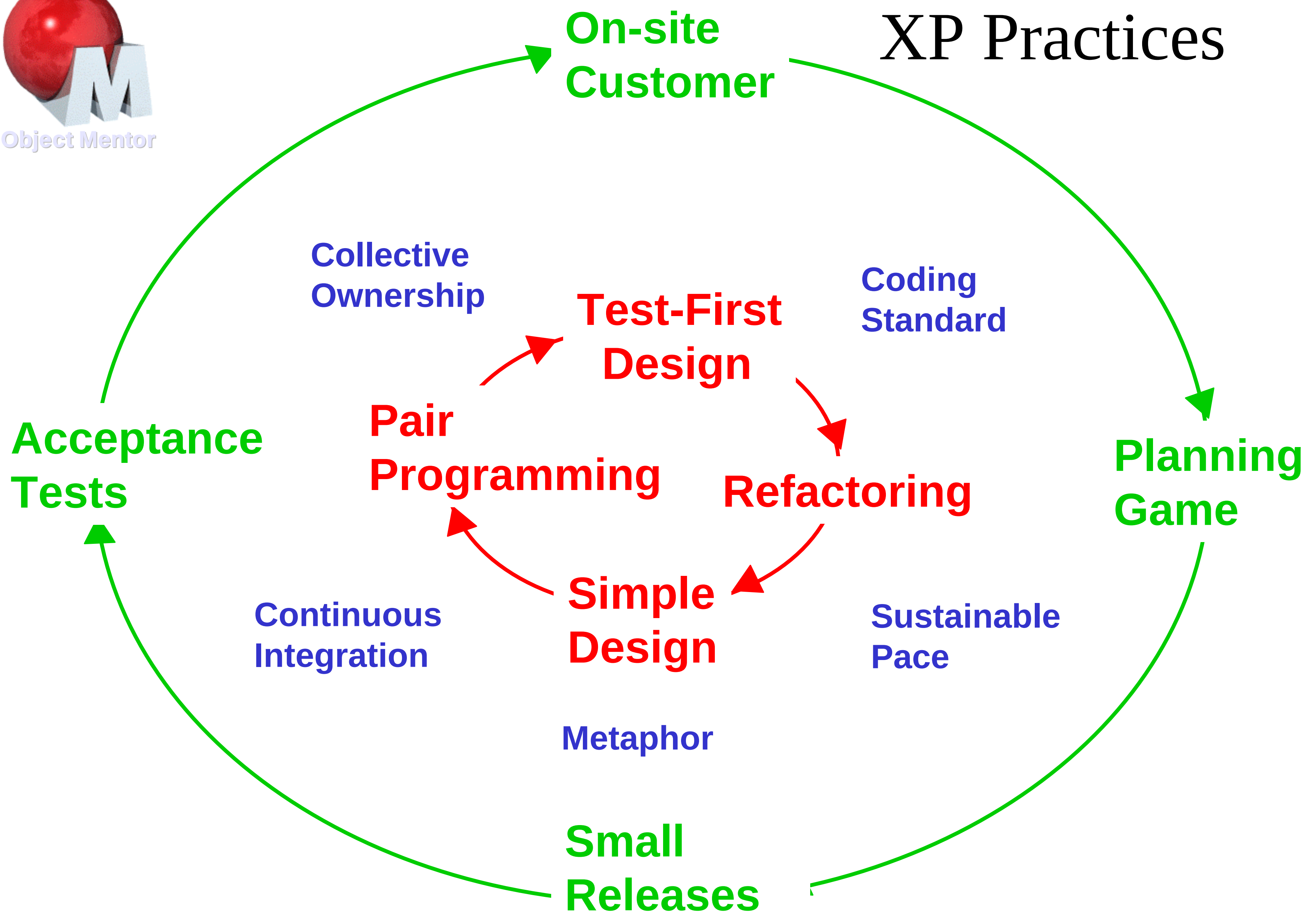
- An agile methodology for developing software:
  - with small-to-medium sized teams
  - with uncertain or rapidly changing requirements
  - at a sustainable rate
- "Agile?"
  - A "lightweight" body of methods for delivering software to customers
  - XP, Scrum, Crystal, DSDM, Adaptive Software Development, Feature-Driven Development

Agile Processes – an alternative

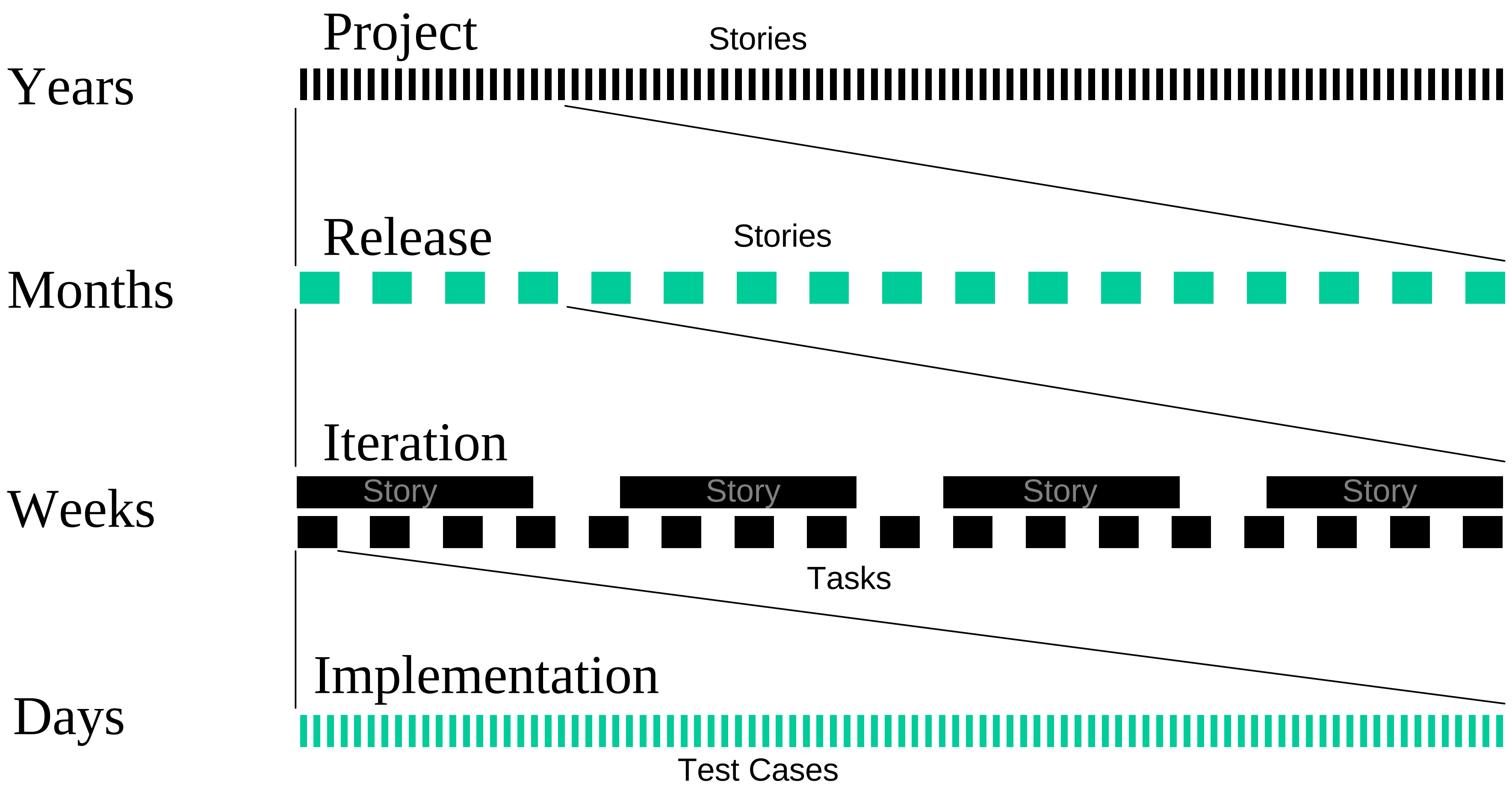
- The Agile Alliance Values:
  - **Individuals and interactions** over process and tools,
  - **Working software** over comprehensive documents,
  - **Customer collaboration** over contract negotiation,
  - **Responding to change** over following a plan.

» *From: "The manifesto for Agile Software Development"*

XP Practices



XP Process Overview



A Discipline...

- XP is a *discipline* of software development
- You *must* do the practices!
- The industry considers certain practices good
- We do them to extreme levels for extreme results
- Examples:
  - Code reviews and design reviews are good
    - Review all the time (Pair Programming)
  - Testing is good
    - Everybody tests all the time
      - Programmers (Unit Testing)
      - Customers (Acceptance Testing)

Object Mentor

Success Story #1

- B2B settlement services
- Multiple, high risk clients
- Web based system
- ASP/Java/Oracle
- 25 person team
- Historically low quality and late
- Split into multiple teams
- Eliminated specialization
- Greenfield development
- Aggressive refactoring
- Short iterations
- Critical for their survival

Object Mentor

Success Story #2

- Shrink Wrapped software
- 3 teams, 2 sites
- Heavy dependencies on outside groups
- Historically low quality and late
- Long test/fix cycles
- Heavy emphasis on incremental planning
- Data collection
- Lots of testing
- Jury still out
- January, 2002 Software Development magazine

Success Story #3

- R&D environment - genetics
- Poor customer relationship
- Development took too long
- Low quality and late

- On site customer
- Quick wins
- Short iterations
- Customers delighted
- Can’t keep up with developers

Object Mentor

References

- www.xprogramming.com
- www.junit.org
- www.pairprogramming.org
- groups.yahoo.com/group/extremeprogramming
- *Extreme Programming Explained*, Kent Beck
- *Planning Extreme Programming*, Kent Beck, Martin Fowler
- *Extreme Programming Installed*, Ron Jeffries, Chet Hendrickson, Ann Anderson