The Agile Alternative

Carl Erickson, PhD
Atomic Object LLC
active, acute, alert, athletic, brisk, buoyant, bustling, clever, deft, dexterous, easy-moving, energetic, fleet, frisky, limber, lithe, lively, mercurial, prompt, quick, quick-witted, rapid, ready, sharp, spirited, sportive, sprightly, spry, stirring, supple, swift, vigorous, vivacious, winged, zippy
Agile Alternative

• Learn enough to start
• Work together in the same space
• Express solution in features
• Maintain working system, fully tested, ready to ship
• Deliver features consistently
• Evolve the design as you learn
Agile as planned

![Graph showing the relationship between time, features, and requirements.](image-url)
Data-driven Projects

How do you know the state of a project?

How can you predict delivery date?

How well does a project cope with change?
Infrastructure first

- Application protocol
- TCP connections
- View framework
- Data source
- View framework
- Controller
- GUI widgets
- Security
- Driver view
- Drive curve
- Playback mode
- Vehicle build
Features first

Application protocol
TCP connections
View framework
Data source
View framework
Controller
GUI widgets
Security
Driver view
Drive curve
Playback mode
Vehicle build

(time)
Measure and Predict
Data-driven projects

![Burn Down Chart](image)

Generated on: January 3, 2006

- Project Burndown
- Points
- Iteration
- Completion Date
Making adjustments

- Requirements
- Infrastructure
Extreme Programming?

2am, marathon, Cheetos, hacking, greasy, undocumented, brilliant, tests?, mysterious, Jolt Cola, protective, ad-hoc, brittle, cubicle, fear change, cowboy, overly elegant, code ego, real soon now, stress difficult to talk to
Extreme Programming!

2am, marathon, Cheetos, hacking, greasy, undocumented, brilliant, tests?, mysterious, Jolt Cola, protective, ad-hoc, brittle, cubicle, fear change, cowboy, overly elegant, code ego, real soon now, stress difficult to talk to

8-5, teamwork, code reviews, communication, coding standards, snacks, communal code tests, tests, tests, maintainable, documented predictable, copes with change
XP Practices

Whole Team

Coding Standard

Test-Driven Development

Refactoring

Sustainable Pace

Simple Design

Metaphor

Small Releases

Pair Programming

Continuous Integration

Collective Ownership

Customer Tests

Planning Game
Core of Agile

Customer-driven iterations
Test-driven development
One-big-room, collaboration
Continuously working system
Iteration Ritual

- Demo, feedback, questions, priorities, planning
- Programming, testing
- Publish build, Reflection, Estimation
Customer-driven Iterations
Acceptance Test Automation
as_commissioner do
  connect_to_draft
  turn_autoselect_off
  wait_for_draft_to_start
  pause_draft_now
  change_pick_time_to 17.seconds
  wait 30.seconds
  pause_draft_now
  modify_pick 'R1P2' do
    set_player_to 'Peppers, Julius'
  end
  resume_draft
end

create_draft
  name 'test draft'
  starts now + 1.minutes
  sport 'football'
  num_rounds 15
  num_teams 6

  draft_type 'live'
  trading_allowed false
  draft_order 'serpentine'

  draft_signup
    add_human_teams 6
    add_ai_teams 0
Test-driven Development

1. Write a test, run it, watch it fail.
2. Write the simplest possible code to pass the test.
3. Run it, watch it pass, refactor as necessary.

(Automation is necessary, not very difficult.)
TDD Impact

10x
One Big Room & Pairs
Day in the Office
Continuous Integration
Continuously Working System

• Binary status
  Done, not started
• Breaking the build
  Finding bugs faster
  Big bang problems
• Change in business
  Budgets
  Priorities
  Scope
• Feedback
  Clients
  Users

January 2006
Economic Implications

Host Controller

- **Dynamometer Fan Sequencer Drivers Aid**
- **Data Acquisition Throttle Controller**
- **Full Robot**

- Blue line: Classical Project
- Red line: Incremental Release

Revenue vs. Months
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck  James Grenning  Robert C. Martin
Mike Beedle  Jim Highsmith  Steve Mellor
Arie van Bennekum  Andrew Hunt  Ken Schwaber
Alistair Cockburn  Ron Jeffries  Jeff Sutherland
Ward Cunningham  Jon Kern  Dave Thomas
Martin Fowler  Brian Marick
Adopting Agile

• Do so with agility
• Know you’ll need to adapt
• Find a supportive customer
• Find a coach to help