



## *Confessions of a Former Agile Naysayer*

***SSTC 2007***

Arlene F. Minkiewicz

Chief Scientist

PRICE Systems

[arlene.minkiewicz@pricesystems.com](mailto:arlene.minkiewicz@pricesystems.com)

- Setting the stage
- What exactly is agile development?
- Our implementation of agile
- Unique agile challenges at PRICE
- Agile Successes and Lessons Learned
- Conclusions

- Two years ago our software organization was producing good software but we....
  - weren't always building the right software
  - were struggling with team communication and cohesion
  - were struggling with communicating software successes to the organization
  - were attempting a project significantly bigger and more complex than any we had tried before
  - were looking to improve our processes for gathering and recording customer requirements
- At the time we had just separated requirements work from other software development work (organizationally)
  - We were starting to see successes with this
- Our Product Development Director returned from a trade show convinced that the answer to all our software woes was the Agile Development Process

- I was skeptical.....
  - We had just struggled with process changes and now we weren't giving them enough time to work
  - The requirements work wasn't a slam dunk but we were making progress
  - We had just grown a group of business analysts devoted to developing formal requirements and now we were changing to a process that had no formal requirements
  - Although we were a small software development company, well suited to an agile deployment – the type and complexity of the software that we build did not appear to lend itself well to an agile process
  
- But I was wrong...
  - While not a panacea, agile definitely has had positive impacts on our development process
  - Agile did help with internal and external communication issues
  - Agile did help bring our software team together
  - Agile did help us develop customer pleasing software
  - There were other benefits as well.....

## *What Exactly is Agile Development?*

---

- According to the Merriam-Webster on-line dictionary “agile” means:
  - “1: marked by ready ability to move with quick easy grace;”
  - “2: having a quick resourceful and adaptable character.”
- In agile software development, “agile” tends to mean “the ability to respond to change.”

- Empowered, self-organizing teams
- Multi-discipline, cross-functional teams (whole team culture)
- Project- and product-centric focus, minimal organizational focus
- Shared responsibility, role-based accountability
- Shared vision of standards of excellence
- Close, continuous collaboration, direct communication

- Early, frequent, and continuous demonstration of progress through concrete deliverables
- Rapid feedback, reflection, learning, adjustment
- Small work batch sizes, minimal specialization, reduced queuing delays
- Just in time production, minimize production of artifacts not immediately (or ever) consumed
- Low friction – simplicity, minimalism, pragmatism
- Avoidance of “debt,” focus on forward movement
- Parallelism and opportunistic control
- Sustainable, constant, predictable pace



## *Agile Values*

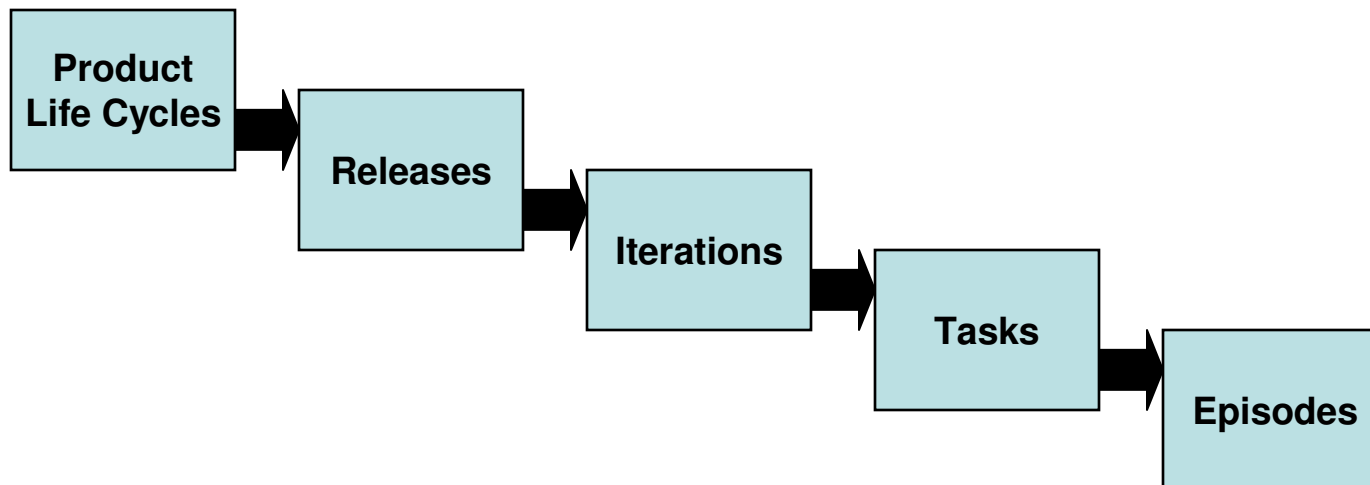
---

- Communication
- Simplicity
- Feedback
- Courage



- Rapid Feedback
- Assume Simplicity
- Incremental Change
- Embracing Change
- Quality Work
- Teach Learning
- Small Initial Investment
- Play to Win
- Concrete Experiments
- Open Honest Communication
- Work With Instincts
- Accepted Responsibility
- Local Adaptation
- Travel Light
- Honest Measurement

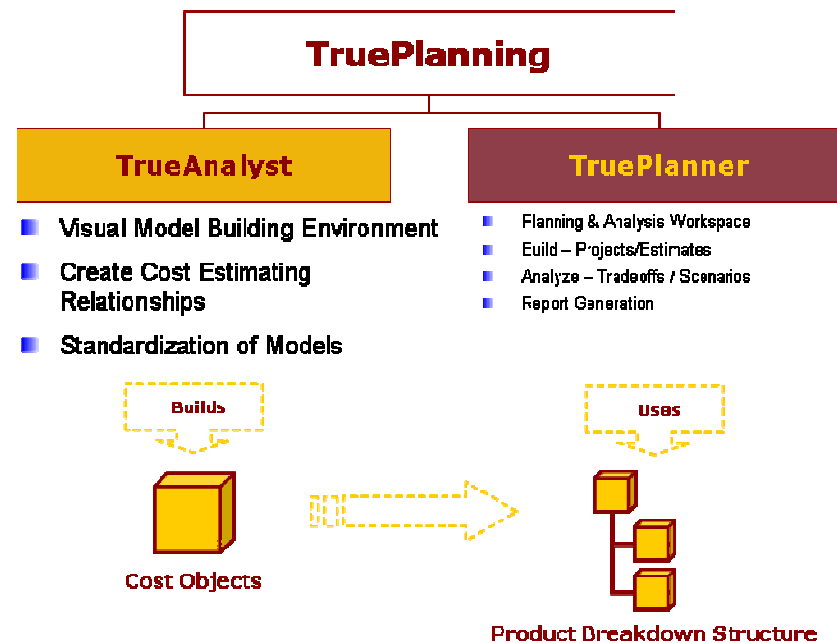
- Agile is iterative and incremental
- Agile is driven by time-boxed cycles
- The rhythm of the Agile process is crucial



- Project Management
- Test Driven Development
- Simple Design
- Pair Programming
- Refactoring
- Continuous Integration
- Collective Ownership
- User Stories
- Project Management

- **Project Management**
  - Three year plan for themes (high level objectives)
  - One year plan for features (the next level pieces of the themes)
  - Features broken down into User Stories that are achievable in chunks of less than 2 weeks
  
- **Business Case Analysis**
  - Developers and business analysts collaborate to estimate stories
  - Business Analysts Prioritize Stories
  - Top priority stories are selected for next iteration
  - Tests are developed to define success for each story
  
- **Software Development**
  - Developers accept stories in priority order – discouraging specialization
  - Pair programming for all production code
  - Write enough code to make the tests pass
  - Regression tests to ensure existing functionality persists
  - On-going refactoring

- Two distinct aspects of Product Development
  - Traditional Software
  - Cost Research
- Cost Research activities
  - Data Collection and Analysis
  - Development of Cost Estimating Relationships
  - Implementation
  - Regression testing
- Cost Research Implementation via TrueAnalyst SW product
- Cost Research Testing via TruePlanner SW Product
- How to model non traditional activities and competing development priorities within agile??



- **Project Management**
  - Stories identified as Cost Research or Product Development
  - Data collection and analysis activities broken into two week chunks
    - Multiple stories for data collection efforts that require more time - further stratifying criteria as more is learned
    - Sometimes just have to time box
  - Balance of competing priorities when software development necessary for further Cost Research Implementation
  
- **Business Case Analysis**
  - Cost Researchers collaborate with Business Analysts to estimate stories
  - Cost Researchers collaborate with Business Analysts to supply test data
  
- **Cost Researchers**
  - Cost Researchers accepts stories in priority order
    - Harder to discourage specialization since any cost discipline requires substantial subject matter expertise.
  - Pair programming only when it makes sense
  - Use regression tests to ensure cost changes in one area don't cause undesired costs because of interrelationships

- External Communication
  - Bi-weekly walk throughs keep organization abreast of progress and helps set realistic expectations
  - Published statistics facilitates negotiations as requirements change
- Internal communications
  - Planning meetings
  - Task breakdowns
  - Exchange of ideas and challenges
- Team Cohesiveness
  - Developers talking every day
  - Effective Interaction between Software Development and Cost Research
- Quick turn around with new hires
  - Pair programming
  - Everyone familiar with all functionality – there’s always someone to ask
- Improved Quality
  - Daily builds with automated testing
  - Regression testing
  - Early detection of issues

## *But there are still challenges...*

- Continuous integrations sometimes happen too quickly
  - Software changes or cost research changes that caused test to break?
  - How do we get the tests green with so many things changing at once
  - Cost changes are sometimes desired but keeping the tests up to date is a challenge
- Simplicity
  - Hard to do 'only what's needed'
  - Still overcomplicating some things
- Focus
  - Culture of interruption
  - Finding the right balance between thinking and collaborating
- Still struggling with breaking down open ended Cost Research tasks
- Some changes don't fit well
  - Architectural and data base changes that have wide impact sometimes have impacts that go beyond the two week iterations



- Original skepticism about adapting agile practices was overrated
- Agile – even with adaptations for our unique situation – had positive impacts
  - Improved communication both internal and external
  - Getting and incorporating feedback from internal and external customers
  - Improved flexibility, reduced resistance to change
  - Improved quality
  - Improved team dynamic
  - Quicker reaction to team changes
- But not a panacea
  - Culture change was not easy
  - Unique aspects of our development presented challenges not really covered in the text books
  - Not everything lends itself to agile processes

- **Leader in Program Affordability Management solutions**
- **Combine cost estimating, project control, and knowledge management – *ensuring project success at every decision gateway***
- **Customers increase visibility, minimize risk and cost, accelerate project development, and improve the effectiveness of project selection, control and delivery**

[www.PRICESystems.com](http://www.PRICESystems.com)



POWERED BY  
**PRICE** Research