

# Beyond the Customer: Agile Business Practices for XP

XP Agile Universe 2002

Paul Hodgetts

President, Agile Logic Inc.

**Agile Logic**

*Innovation in Software Development*

Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL, Aug 4-7, 2002

# Today's Plan

- Introduction
- Agile Product Development
- Traceability
- Multi-Customer Strategies and Practices
- Generating Stories from Product Strategies and Features
- Issues in Adopting Agile Business Practices

# Introduction

Welcome!

**Agile Logic**

*Innovation in Software Development*

Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL, Aug 4-7, 2002

# Your Presenter

- Your presenter is Paul Hodgetts
- Founder and president of Agile Logic, based in Southern California
- Agile Logic provides training, coaching, consulting, and custom development services
- 19 years experience in all aspects of software development
- Served as coach, developer, and mentor for agile teams for the past 2½ years
- Recent focus on agile processes, enterprise Java, and organizational change management
- Contributing author to Extreme Programming Perspectives
- Previous presenter at XP Universe and JavaOne
- Taught at C.S.U.F, member of XP, Java, J2EE, C++ advisory boards

# Why Agile Business Practices?

- We're not real successful at software development
- The same issues affect all of the product development process
- Sometimes the business side of development is more ad hoc than the development side!
- Development needs good business side practices for success

# The Defined, Controlling/Optimizing/Repeatable Model Doesn't Work So Well

- Software development is not really manufacturing
- Software development is not really engineering

# An Empirical, Adaptive Model Works Better

- Software development is more of a craft
- Software development is “a cooperative game of invention and communication”

Alistair Cockburn

# Goals of Agility

- Delivering the most value to the business, efficient use of resources, maximize ROI and time-to-ROI
- Faster development, higher productivity
- Flexibility to respond to change and leverage learning
- Better quality solutions, more enduring systems
- More fulfilling development culture



# Characteristics of Agility

- 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

# Characteristics of Agility

- 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

# XP and the Development Pipeline

- XP is oriented towards development, and the interface to development
- When adopted and effectively practiced, results are a high-performance development team
- XP needs to be fed with business and product strategies

# What Goes Wrong Between the Business and Development Sides

- Sometimes the business side can't feed development fast enough
- Sometimes the business cycles don't match with shorter release and iteration cycles
- Sometimes the business folks are not actively collaborating with the development folks
- Sometimes the business folks base decisions on less than concrete, empirical data

# The Business Side Can Be Agile, Too!

- Incorporate all involved folks into a collaborative Whole Team
- Organize business and product strategy-setting into agile cycles
- The business and development sides need to spin at compatible rates
- Base business and product strategies on concrete, testable goals
- Be incremental – avoid large-artifact, up-front activities

# Agile Product Development

**Agile Logic**

*Innovation in Software Development*

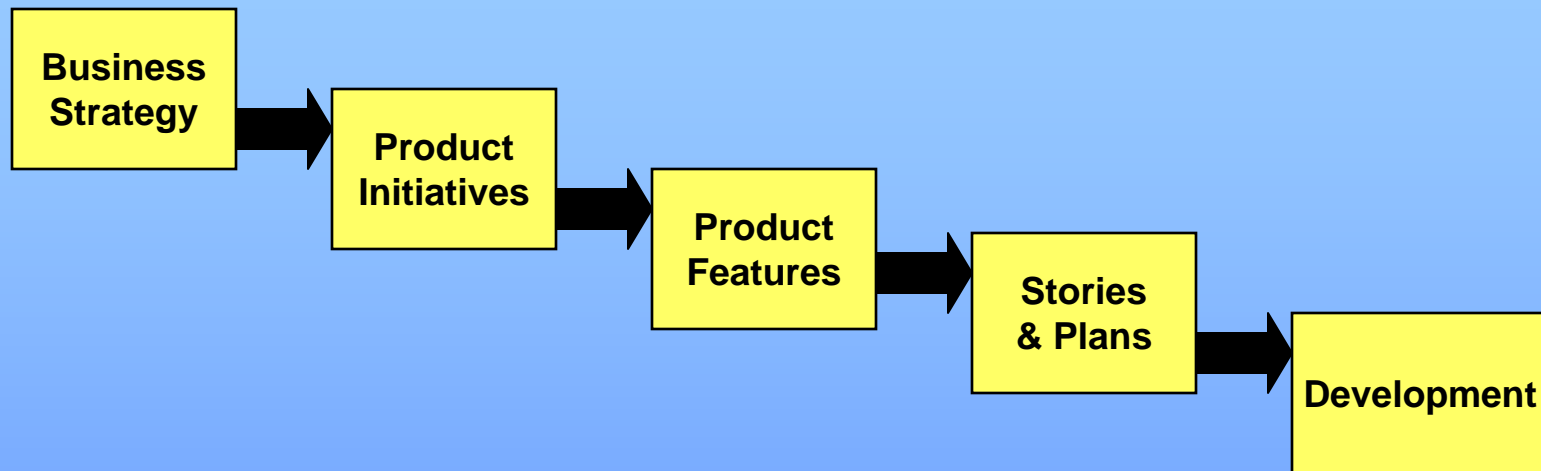
Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL Aug 4-7, 2002

# The Scope of Product Development

- Overall Business Objectives
- Product Initiatives
- Product Feature Sets
- Stories and Plans
- Development
- XP offers specific practices for the last two, assumes the first three occur

# Product Development Pipeline





# Overall Business Objectives

- What is the business trying to accomplish?
- Defined in terms of revenue, growth, market share...
- Based on satisfying the needs of the business stakeholders
- Done by board of directors and/or executive management

# Product Initiatives

- What kind of product development supports business objectives?
- Defined in terms of overall products
- Products targeted at market segments, general types of users, contracts...
- Based on industry and market research
- Done by marketing, perhaps with executive management

# Product Feature Sets

- What does the product have to deliver to realize the initiatives?
- Defined in terms of features and functionality
- Features targeted at the needs of the identified users
- Based on market research, user workshops, competitive analysis...
- Done by marketing and/or product development

# Stories and Plans

- How will we go about implementing the features?
- Defined in terms of stories and release plans
- Targeted at the user needs, balanced by business needs and schedules
- Based on market windows, development capabilities (velocity)
- Done by product management, project management, development

# Development

- Make it so using XP development practices
- Results in:
  - Iteration Plans
  - Acceptance tests
  - Tasks
  - Unit tests
  - Code
  - Deliverable Product

# Business Cycles and XP Cycles

- Wrapping the XP cycles with the business cycle
- Business cycle provides raw materials for XP cycles

# XP Cycles

- Programming episode
- Story/task cycle
- Iteration cycle
- Release cycle

# Business Cycle Practices

- Cycle Charter
- Product Strategy Planning
- Product Feature Set Planning
- Release Planning and Strategic Feedback
- Strategic Investment and Resource Commitments
- Release Development
- Business Retrospectives



# Cycle Charter

- Defines overall business objectives
- Business objectives defined with business stories
- Business story states a testable business goal
- Business stories may not be “completed”
- Example: Increase revenue by 15%
- Business stories can also define overall constraints
- Example: Development costs can not exceed \$1.2M per month

# Product Strategy Planning

- Defines products and capabilities
- Products defined with product stories
- Product stories can be tested, but are generally an intermediate artifact
- Example: Sell a shopping cart system for retail book sellers

# Product Feature Set Planning

- Defines product features, requirements, priorities
- Product features defined with informal, light-weight use cases
- Use case incrementally developed and tested with stories
- Use cases can be an intermediate artifact, but useful for communication
- Example: Use case for “User adds book to shopping cart”

# Release Planning and Strategic Feedback

- Defines stories for implementing use cases
- Produces prioritized and estimated story queue
- Standard XP release planning practices
- Results of planning used as feedback for investment planning

# Strategic Investment and Resource Commitments

- Defines resource constraints (team size, equipment, budget) for cycle
- May loop back to adjust plans based on resource constraints
- Resource adjustments may take time (hiring, purchasing)

# Release Development

- Standard XP iteration development cycles
- Iteration results may force adjustments to business cycle
- Release retrospective done to provide business feedback

# Business Retrospectives

- Release results used to learn and adjust business cycle
- Test the business stories
- Business retrospective done to provide feedback for next cycle

# Business Cycle Duration

- Business is a bigger ship and turns more slowly
- Feedback takes longer to measure revenue, market share changes
- Minimize cycle duration – quarterly, semi-annually at most
- Schedule multiple releases per cycle where possible



# Business Cycle Intervals

- Business cycles may be offset from XP cycles
- Business planning has lead time and needs to be ready for releases
- Business retrospective needs delay to measure slow-moving feedback
- No easy answer, but a rhythm will emerge

# Concrete Business Feedback

- Avoiding guesswork and pet projects
- Measurement with real metrics
- This is business, not science – keep it simple!
- Metric resolution should be realistic with acceptable tolerances

# Example of Concrete Feedback

- Business Story: Increase revenue by 15%
- Metric: Measure revenue, calculate increase

# Example of Concrete Feedback

- Product Story: Generate 10% more revenue from shopping cart product
- Product Feature Set Planning: Reduce number of cancelled transactions by 20%
- Use Case: Change shopping cart check out to ask for just-in-time registration info
- Metric: Usability testing to validate new use case is less threatening
- Metric: Measure number of cancelled transactions
- Metric: Measure revenue from shopping cart product, calculate increase

# Traceability in the Overall Process

# Why Bother with Traceability?

- Businesses waste effort with inefficiencies and fuzzy strategies
- Better-focused development can provide a competitive advantage
- Better defined and communicated strategies creates purposed teams
- Teams with a collective purpose are more motivated

# Misusing Traceability

- Traceability is not an accountability or blame mechanism
- Traceability is not an enforcement mechanism for command and control
- Traceability is not an after-the-fact way to validate strategies
- Traceability should be obvious and not need verification practices

# Forward-Backwards Traceability

- Traceability flows naturally forward when using generative practices
- Traceability is best demonstrated by looking backwards
- “Jane is working on a task to help increase our quarterly revenue by 15%”



# The Traceability Chain

- Development episodes trace back to tasks
- Tasks trace back to release/iteration plans and stories
- Stories trace back to product feature plans (use cases)
- Features trace back to product strategy stories
- Product strategies trace back to business stories

# Multi-Customer Strategies and Practices

**Agile Logic**

*Innovation in Software Development*

Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL, Aug 4-7, 2002

# The Customer

- In XP, the Customer works directly with the Developers
- In Scrum, the Customer is called the Product Owner
- Canonical XP has the End User fulfilling the Customer role
- The Customer defines requirements and priorities

# The Distant Customer

- What if there is space and/or time distance between the End User and Developers?
- Employ a User Proxy as End User representative to the Developers
- The User Proxy fulfills the XP Customer role

# Many Customers

- What if there are multiple End Users?
- Form a Customer Board whose constituents represent the End Users
- The Customer Board appoints a single User Proxy as representative to the Developers
- The User Proxy has the accountability as Customer
- Multiple members of the Customer Board may work with the Developers

# Diversity of Customers

- What if there are multiple types of End Users and/or multiple Stakeholder interests?
- Stakeholders are not End Users, but have a say in requirements and priorities
- The Customer Board pattern still applies
- Each member's voting interest may be weighted

# Customer Practices for Customer Boards

- Establish the Customer Board
- Story Telling
- Mapping and Merging Stories
- Organizing Stories to Strategic Priorities
- Establish Normative Priorities

Thanks and credit to Brad Appleton for his XP list posting!

# Establish the Customer Board

- Choose members based on their ability to represent their interests
- Determine the number and composition of the board
- Number of members based on covering the varying types of interests
- Weighting of members based on importance of their interests to the overall product
- Board composition should be subject to retrospectives and adjustments



# Story Telling

- Each board member has their requirements and priorities
- Allow each member to complete their release stories one by one
- All other board members are expected to actively listen
- Use different color cards for each board member

# Mapping and Merging Stories

- Lay out all stories on the table
- Map duplicate stories to a single story
- Merge similar stories into a single story
- Keep placeholders for mapped and merged stories

# Organizing Stories to Strategic Priorities

- Identify stories that fulfill strategic business and product strategies
- Assign a priority to each story based on this mapping
- Story priority is usually the higher of the applicable strategic priorities
- Group stories into strategic priority categories

# Establish Normative Priorities

- Go through each strategic priority category
- Each story in category is assigned a normative priority
- Normative priority is based on the board member's weighting and story priority

# Forming a Single Release Plan

- Review the overall layout of the stories by strategic category and normative priority
- Decide on allocation of release plan bandwidth to strategic categories
- Fill single release plan stream with stories
- Adjust as necessary based on group consensus
- The Customer is accountable for representing the results

# Generating Stories from Product Strategies and Features

# Generating Product Strategies and Features

- Not a formulaic operation – requires some leg work
- Based on product development best practices
  - Market research
  - Competitive analysis
  - Customer feedback
  - Usability testing

# Product Feature Set Generation

- Canonical XP is incremental and just-in-time
  - Need a car... A convertible... A red one... With under 15,000 miles...
- But many cultures are more up-front and carry inventory
  - Larger plans required for funding
  - Contracts specify “fixed” scope



# Documenting Feature Requirements

- Use best practices – use cases
- But don't over-do it
  - Cockburn's lighter-weight format
  - Constantine's essential use cases
- Overall use cases useful for generating a shared mental model
- Fill in detail as incrementally as possible

# Use Cases vs. Stories

- A story is not a use case, a use case is not a story
- A use case defines functional requirements in total
- Defines breadth and depth of system behavior
- Additional non-functional requirements often needed
- A story defines a piece of system capability to be implemented
- Stories as change requests – add, modify or remove capability

# Extracting Stories from Use Cases

- The overall set of use cases is the quilt
- The stories are the patches that are incrementally sewn together to fill it in
- Story scope – what use case(s) are being asked for?
- Story breadth – what use case scenarios are being asked for?
- Story depth – what level of completion is being asked for?

# Example of Use Cases and Stories

- Overall set of use cases for shopping cart system

# Example of Use Cases and Stories

- Example of story that maps to use case
- “Implement the User Checkout use case. Handle only the positive path, credit card payment scenario. Don’t worry about sales tax or discount calculations.”

# Example of Use Cases and Stories

- Example of a cross-cutting story
- “Implement sales tax calculations. Sales tax calculations are needed in these use cases: Review Shopping Cart, User Checkout, Admin Review Order. Handle only a single fixed sales tax rate – don’t worry about state tax tables.”

# Issues in Adopting Agile Business Practices

**Agile Logic**

*Innovation in Software Development*

Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL, Aug 4-7, 2002

# Issues in Adopting Agile Business Practices

- The Value of Leaders and Mentors
- A Bit of Preparation and Education Never Hurts
- Respect Everyone's Turf – This Isn't Just a Programmer Thing
- Countering Fear and Resistance One Small Step at a Time



# Issues in Adopting Agile Business Practices

- Pilot Projects and Adoption on a Shoestring
- The Challenge of Building and Sustaining Momentum
- The Fall of Empires and the Rise of the Collective
- Adapting and the Learning Organization

# Want More?

- Whitepapers coming from this presentation
- Whitepapers to be organized and expanded into a book
- Agile Logic can help you implement what you've learned in this tutorial
- Paul can be contacted at:
  - Paul Hodgetts
  - Agile Logic
  - 1519 East Chapman Ave. #254
  - Fullerton, CA, USA 92831
  - Phone: (714) 269-7401
  - Email: [phodgetts@agilelogic.com](mailto:phodgetts@agilelogic.com)
  - Website: [www.agilelogic.com](http://www.agilelogic.com)

# Thank You for Attending!

## Enjoy the rest of the Conference!

**Agile Logic**

*Innovation in Software Development*

Copyright © 2002 Paul R. Hodgetts All Rights Reserved

**XP Agile Universe**  
Chicago, IL, Aug 4-7, 2002