AGILE DEVELOPMENT AND ITS IMPACT ON PRODUCTIVITY

2006
International Software Measurement & Analysis Conference

David Garmus

The David Consulting Group
Achieving Software Excellence

www.davidconsultinggroup.com
Topics

• Characteristics of Agile Projects
• Performance Data on Agile Projects
• When to Choose Agile
Topics

- Characteristics of Agile Projects
  - Common View
  - Core Practices
  - Life-cycle Development
- Performance Data on Agile Projects
- When to Choose Agile
Agile View – Rigid methods have one thing in common, too much is planned in an uncertain environment at project inception.

“‘XP’ (most common agile methodology) is a lightweight methodology for small-to-medium-sized teams developing software in the face of vague or rapidly changing requirements.”

Kent Beck
## Characteristics Overview

<table>
<thead>
<tr>
<th>Barry Boehm and Richard Turner</th>
<th>Agile</th>
<th>Planned (Traditional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Changeable</td>
<td>Larger Teams &amp; Projects</td>
</tr>
<tr>
<td>Management</td>
<td>Customer Part of Team</td>
<td>Document Plans &amp; Requirements</td>
</tr>
<tr>
<td>Technical</td>
<td>Short Increments Many Releases</td>
<td>Voluminous Test Cases and Plans</td>
</tr>
<tr>
<td>Personnel</td>
<td>Highly Capable Thrive on Chaos</td>
<td>Specialists who Thrive on Order</td>
</tr>
</tbody>
</table>
Core Practices

• Talent & Skill (Fewer, but Better)
• Active Stakeholder Participation
• Assimilate Change Easily (Plan & Re-plan)
  – Incremental Development
  – Use of Simple Models to Address Requirements
  – Rapid Feedback (Just-in-time Requirements & Design)
  – Frequent Delivery
  – Different Strategies for Different Projects
• Steady Development Rate
• High Quality in Work
• Direct Communication (Daily Scrum, Less Paper)
• Maximize Agility through Model Reuse
• Team in Close Proximity
Agile Manifesto (2001)

We 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
Life-cycle Development

Requirements, Design and Coding

• Small in Scope
  –Less than 200 Function Points
  –Many less than 100 Function Points
• High Level Business Requirements
• Lack of Formal Documentation
  –System Descriptions
  –Technical Design Documentation
  –Process Models
  –System Architectures
• Team Development
  –Small, Cohesive, Same Work Space
  –Self Organized and Motivated
  –Driven by Team Dynamics vice Formal Process
Life-cycle Development

Testing and Implementation

- Effort Not Separate for each Testing Type/Phase
- Testing Occurs Concurrently
- Testing by Developers
- Smaller Deliverables, More Releases
Topics

• Characteristics of Agile Projects
  – Common Views
  – Core Practices
  – Life-cycle Development

• Performance Data on Agile Projects
• When to Choose Agile
Use Measurement To Enable Comparisons Between Methods

QUANTITATIVE

- Deliverable Size
- Effort/Cost
- Duration
- Quality

Measurable Performance

QUALITATIVE

- Process
- Methods
- Skills
- Tools
- Environment

Capability Maturity

Baseline of Performance

Measure how you are doing

Identify what you are doing

Standard of performance
Utilize Measurement Results In Decision Making

- Improvements resulting from current and future initiatives must be measured

- The basis for measuring improvements may include:
  - Industry data
  - Organizational baseline data

- It is necessary for the organization to put a “stake in the ground” relative to current performance level in order to improve development practices
## DCG Data Base

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Complexity Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logical Algorithms</td>
</tr>
<tr>
<td>Project Type</td>
<td>Mathematical Algorithms</td>
</tr>
<tr>
<td></td>
<td>Data Relationships</td>
</tr>
<tr>
<td>Platform</td>
<td>Functional Size</td>
</tr>
<tr>
<td>Data Base</td>
<td>Reuse</td>
</tr>
<tr>
<td>Method</td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>Code Structure</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
</tr>
<tr>
<td></td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td>Warranty</td>
</tr>
</tbody>
</table>

## Metrics

- Size
- Cost
- Effort
- Duration
- Defects

## Attributes

- Management
- Definition
- Design
- Build
- Test
- Environment

- Process
- Skill Levels
- Quality Practices
- Measures
Collecting & Reporting

• Identify data set (typically project oriented)
• Collect baseline data
  • Project measures (e.g., effort, size, cost, duration, defects)
  • Project attributes (e.g., skill levels, tools, process, etc.)
• Analyze data
  • Performance comparisons (identification of process strengths and weaknesses)
  • Industry averages and best practices
  • Performance modeling (identify high impact areas)
• Report results
Using Historical Delivery Rates

**DEFINITION**
- REQUIREMENT
- PROJECT SIZE and COMPLEXITY

**CAPABILITY**
- FUNCTION POINT SIZE
- RATE OF DELIVERY

**EFFORT**
- HOURS per FUNCTION POINT
- Schedule
- Effort
- Costs
### Average Hours/Function Point of Recent Enhancement Projects Across Different Platforms from DCG Database for Small Projects

<table>
<thead>
<tr>
<th>Platform</th>
<th>Traditional</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Server</td>
<td>6.5</td>
<td>4.2</td>
</tr>
<tr>
<td>Main Frame</td>
<td>8.1</td>
<td>7.0</td>
</tr>
<tr>
<td>Web</td>
<td>4.8</td>
<td>3.2</td>
</tr>
<tr>
<td>e-business Web</td>
<td>6.6</td>
<td>5.8</td>
</tr>
</tbody>
</table>
Topics

• Characteristics of Agile Projects
  – Common Views
  – Core Practices
  – Life-cycle Development
• Performance Data on Agile Projects
• When to Choose Agile
Which Methodology Should I Use?

- **Waterfall, Iterative and Spiral Methods**
  - Predictive Performance
  - Large Teams
  - Highly Structured Environments
  - Outsourced or Multi-sourced Projects
  - High Financial or Safety Risk
  - Significant Hardware Integration

- **Agile Methods**
  - Exploratory Projects
  - Small Teams
  - Participative Environments
    - Experienced Personnel
    - Active Business Partners
  - Software Dominant Projects
  - In-sourced Projects
  - High Risk of Unknown Requirements
Do I Choose Agile or a Hybrid?

• Initial arguments for selecting a hybrid (traditional) method.
  – High level of risk
  – Large size of project
  – Specified delivery commitment
  – Organizational environment

• Suggested selection process:
  – Map agile attributes based on organization’s tolerance for risk and change
  – Some agile practices can be transplanted to another methodology
  – Leverage best practice processes to augment method chosen
Key Attributes of Agile Projects

- The Planning Game
- On-site Customer / User
- System Metaphor
- Simple Design
- Collective Code Ownership
- Coding Standards
- Pair Programming
- Continuous Testing
- Continuous Integration
- Small Releases
- 40 Hour Work Week

Key Attributes of Current Agile Methods

➤ Well Defined
➤ Require Significant Discipline
Example: Team Size as a Driver for Hybrid

Small Team
2 People
- The Planning Game
- On-site Customer / User
- System Metaphor
- Simple Design
- Collective Code Ownership
- Coding Standards
- Pair Programming
- Continuous Testing
- Continuous Integration
- Small Releases
- 40 Hour Work Week

Medium Team
12 People

Large Team
13 or More

Agile

Rigid
Example: Financial Risk as a Driver for Hybrid

Low Financial Risk
No Impact to Accounts

Large Financial Risk
Impact to Accounts

Agile

The Planning Game
On-site Customer / User
System Metaphor
Simple Design
Collective Code Ownership
Coding Standards
Pair Programming
Continuous Testing
Continuous Integration
Small Releases
40 Hour Work Week

Rigid
Conclusions

• Use of Agile methods affects performance outcomes
• Choosing the appropriate methodology will maximize your delivery performance
• Agile performance can be successful
Contact Information

David Consulting Group web site: 
www.davidconsultinggroup.com

David Garmus
dg@davidconsultinggroup.com