Author Background

- Gov’t contractor with **30+ years of IT experience**
- Large gov’t projects in U.S., Far/Mid-East, & Europe

→ Published six books & numerous journal articles
→ Adjunct at George Wash, UMBC, UMUC, Argosy
→ **Agile Program Management & Lean Development**
→ Specializes in metrics, models, & cost engineering
→ Six Sigma, CMMI, ISO 9001, DoDAF, & DoD 5000
→ Cloud Computing, SOA, Web Services, FOSS, etc.
Today’s Whirlwind Environment

- Overruns
- Attrition
- Escalation
- Runaways
- Cancellation

- Global Competition
- Demanding Customers
- Organization Downsizing

- Vague Requirements
- Reduced IT Budgets
- Overburdening Legacy Systems

- Technology Change
- Redundant Data Centers
- Poor IT Security

- System Complexity
- 81 Month Cycle Times
- Lack of Interoperability

- Work Life Imbalance
- Inefficiency
- High O&M
- Lower DoQ
- Vulnerable
- N-M Breach

- Organization Downsizing

References:


Global Project Failures

- Challenged and failed projects hover at 67%
- Big projects fail more often, which is 5% to 10%
- Of $1.7T spent on IT projects, over $858B were lost

Requirements Defects & Waste

- **Requirements defects** are #1 reason projects fail
- Traditional projects specify **too many requirements**
- More than **65% of requirements** are never used at all

**Defects**
- Requirements 47%
- Design 28%
- Implementation 18%
- Other 7%

**Waste**
- Never 45%
- Always 7%
- Often 13%
- Sometimes 16%
- Rarely 19%

What is Agility?

- **A-gil-i-ty** (ə-'ji-lə-tē) Property consisting of *quickness*, *lightness*, and *ease of movement*; **To be very nimble**
  - The ability to create and **respond to change** in order to profit in a turbulent global business environment
  - The ability to **quickly reprioritize** use of resources when requirements, technology, and knowledge shift
  - A very **fast response** to sudden market changes and emerging threats by intensive **customer interaction**
  - Use of **evolutionary**, **incremental**, and **iterative** delivery to converge on an optimal customer solution
  - Maximizing **BUSINESS VALUE** with right sized, just-enough, and just-in-time processes and documentation

What are Agile Methods?

- **People-centric** way to create innovative solutions
- **Product-centric** alternative to documents/process
- **Market-centric** model to maximize business value

<table>
<thead>
<tr>
<th>Customer Collaboration</th>
<th>Contracts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequent comm.</strong></td>
<td><strong>Contract compliance</strong></td>
</tr>
<tr>
<td>Close proximity</td>
<td>Contract deliverables</td>
</tr>
<tr>
<td>Regular meetings</td>
<td>Contract change orders</td>
</tr>
<tr>
<td><strong>Multi-comm. channels</strong></td>
<td><strong>Frequent feedback</strong></td>
</tr>
<tr>
<td><strong>Relationship strength</strong></td>
<td><strong>Manageability/Motivation</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Individuals &amp; Interactions</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Leadership</strong></td>
<td><strong>Lifecycle compliance</strong></td>
</tr>
<tr>
<td><strong>Boundaries</strong></td>
<td><strong>Process Maturity Level</strong></td>
</tr>
<tr>
<td><strong>Empowerment</strong></td>
<td><strong>Regulatory compliance</strong></td>
</tr>
<tr>
<td><strong>Leadership</strong></td>
<td><strong>Structure</strong></td>
</tr>
<tr>
<td><strong>Boundaries</strong></td>
<td><strong>Manageability/Motivation</strong></td>
</tr>
<tr>
<td><strong>Competence</strong></td>
<td><strong>Manageability/Motivation</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working Systems &amp; Software</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clear objectives</strong></td>
<td><strong>Document deliveries</strong></td>
</tr>
<tr>
<td><strong>Small/feasible scope</strong></td>
<td><strong>Document comments</strong></td>
</tr>
<tr>
<td><strong>Acceptance criteria</strong></td>
<td><strong>Document compliance</strong></td>
</tr>
<tr>
<td><strong>Timeboxed iterations</strong></td>
<td><strong>System flexibility</strong></td>
</tr>
<tr>
<td><strong>Valid operational results</strong></td>
<td><strong>Technology flexibility</strong></td>
</tr>
<tr>
<td><strong>Regular cadence/intervals</strong></td>
<td><strong>Infrastructure flexibility</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responding to Change</th>
<th>Project Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Org. flexibility</strong></td>
<td><strong>Cost Compliance</strong></td>
</tr>
<tr>
<td><strong>Mgt. flexibility</strong></td>
<td><strong>Scope Compliance</strong></td>
</tr>
<tr>
<td><strong>Process flexibility</strong></td>
<td><strong>Schedule Compliance</strong></td>
</tr>
<tr>
<td><strong>System flexibility</strong></td>
<td><strong>System flexibility</strong></td>
</tr>
<tr>
<td><strong>Technology flexibility</strong></td>
<td><strong>Technology flexibility</strong></td>
</tr>
<tr>
<td><strong>Infrastructure flexibility</strong></td>
<td><strong>Infrastructure flexibility</strong></td>
</tr>
</tbody>
</table>

How Agile Works

- Agile **requirements implemented in slices vs. layers**
- User needs with **higher business value** are done first
- Reduces cost & risk while increasing business success

### Agile

- Faster
- Early ROI
- Lower Costs
- Fewer Defects
- Manageable Risk
- Better Performance
- Smaller Attack Surface

### Traditional

- Late •
- No Value •
- Cost Overruns •
- Very Poor Quality •
- Uncontrollable Risk •
- Slowest Performance •
- More Security Incidents •

#### Seven Wastes

1. Rework
2. Motion
3. Waiting
4. Inventory
5. Transportation
6. Overprocessing
7. Overproduction

**MINIMIZES**

- JIT, Just-enough architecture
- Early, in-process system V&V
- Fast continuous improvement
- Scalable to systems of systems
- Maximizes successful outcomes

**MAXIMIZES**

- Myth of perfect architecture
- Late big-bang integration tests
- Year long improvement cycles
- Breaks down on large projects
- Undermines business success

Agile Performance Measurement

- **Burndown**
  - Work (Story, Point, Task) or Effort (Week, Day, Hour)
  - Time Unit (Roadmap, Release, Iteration, Month, Week, Day, Hour, etc.)

- **Cumulative Flow**
  - Work (Story, Point, Task) or Effort (Week, Day, Hour)
  - Time Unit (Roadmap, Release, Iteration, Month, Week, Day, Hour, etc.)

- **Earned Value Management - EVM**
  - Work (Story, Point, Task) or Effort (Week, Day, Hour)
  - Time Unit (Roadmap, Release, Iteration, Month, Week, Day, Hour, etc.)
  - CPI
  - SPI
  - PPC
  - APC

- **Earned Business Value - EBV**
  - Work (Story, Point, Task) or Effort (Week, Day, Hour)
  - Time Unit (Roadmap, Release, Iteration, Month, Week, Day, Hour, etc.)
Agile Cost of Quality (CoQ)

- Agile testing is 10x better than code inspections
- Agile testing is 100x better than traditional testing
- Agile testing is done earlier “and” 1,000x more often

Agile Cost & Benefit Analysis

- Costs based on avg. productivity and quality
- Productivity ranged from 4.7 to 5.9 LOC an hour
- Costs were $588,202 and benefits were $3,930,631

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
<th>Trad. Testing</th>
<th>Agile Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>$\left(\frac{10,000}{5.4436} + \frac{3.945 \times 10 \times 100}{100}\right) \times 100$</td>
<td>$588,202$</td>
<td>$233,152$</td>
</tr>
<tr>
<td>Benefits</td>
<td>$\left(10,000 \times 10.51 - 6,666.67 \times 9 \right) \times 100 - 588,202$</td>
<td>$3,930,631$</td>
<td>$4,275,681$</td>
</tr>
<tr>
<td>B/CR</td>
<td>$\frac{3,930,631}{588,202}$</td>
<td>7:1</td>
<td>18:1</td>
</tr>
<tr>
<td>ROI</td>
<td>$\frac{3,930,631 - 588,202}{588,202} \times 100%$</td>
<td>567%</td>
<td>1,734%</td>
</tr>
<tr>
<td>NPV</td>
<td>$\left(\sum_{i=1}^{5} \frac{3,930,631}{5} \div 1.05^5 \right) - 588,202$</td>
<td>$2,806,654$</td>
<td>$3,469,140$</td>
</tr>
<tr>
<td>BEP</td>
<td>$\frac{588,202}{\left(\frac{4,509,997}{588,202} - 1\right)}$</td>
<td>$88,220$</td>
<td>$12,710$</td>
</tr>
<tr>
<td>ROA</td>
<td>$\frac{\text{NORMSDIST}(2.24) \times 3,930,631 - \text{NORMSDIST}(0.85) \times 588,202 \times \exp(-0.05 \times 5)}{</td>
<td>$3,504,292$</td>
<td>$4,098,159$</td>
</tr>
</tbody>
</table>

\[d_1 = \frac{\ln(\text{Benefits} \div \text{Costs}) + (\text{Rate} + 0.5 \times \text{Risk}^2) \times \text{Years}}{\text{Risk} \times \sqrt{\text{Years}}}\]

\[d_2 = d_1 - \text{Risk} \times \sqrt{\text{Years}}\]

Benefits of Agile Methods

- Analysis of 23 agile vs. 7,500 traditional projects
- Agile projects are 54% better than traditional ones
- Agile has lower costs (61%) and fewer defects (93%)

Agile vs. Traditional Success

- Traditional projects succeed at 50% industry avg.
- Traditional projects are challenged 20% more often
- Agile projects **succeed 3x more and fail 3x less often**

Benefits of Organizational Agility

- Study of 15 agile vs. non-agile Fortune 500 firms
- Based on models to measure organizational agility
- Agile firms outperform non-agile firms by up to 36%

Agile Adoption

- VersionOne found 84% using agile methods today
- Most are using Scrum with several key XP practices
- Lean–Kanban is a rising practice with a 32% adoption
Agile Proliferation

- Number of CSMs have doubled to 200,000 in 2 years
- 558,918 agile jobs for only 121,876 qualified people
- 4.59 jobs available for every agile candidate (5:1)

## Agile Industry Case Studies

- **84%** of worldwide IT projects use agile methods.
- Includes regulated industries, i.e., DoD, FDA, etc.
- Agile now used for safety critical systems, FBI, etc.

<table>
<thead>
<tr>
<th>Industry</th>
<th>Org</th>
<th>Project</th>
<th>Purpose</th>
<th>Size</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic Commerce</td>
<td>Google</td>
<td>Adwords</td>
<td>Advertising</td>
<td>20 teams, 140 people, 5 countries</td>
<td>1,838 User Stories, 6,250 Function Points, 500,000 Lines of Code</td>
</tr>
<tr>
<td>Shrink Wrapped</td>
<td>Primavera</td>
<td>Primavera</td>
<td>Project Management</td>
<td>15 teams, 90 people, Collocated</td>
<td>26,809 User Stories, 91,146 Function Points, 7,291,666 Lines of Code</td>
</tr>
<tr>
<td>Law Enforcement</td>
<td>FBI</td>
<td>Sentinel</td>
<td>Case File Workflow</td>
<td>10 teams, 50 people, Collocated</td>
<td>3,947 User Stories, 13,419 Function Points, 1,073,529 Lines of Code</td>
</tr>
<tr>
<td>U.S. DoD</td>
<td>Stratcom</td>
<td>SKIweb</td>
<td>Knowledge Management</td>
<td>3 teams, 12 people, Collocated</td>
<td>390 User Stories, 1,324 Function Points, 105,958 Lines of Code</td>
</tr>
</tbody>
</table>

Agile ROI at Google

- Google early adopter of agile methods and Scrum
- Google also uses agile testing at enterprise scale
- 15,000 developers run 120 million tests per day

- 440 billion unique users run 37 trillion searches each year
- Single monolithic code tree with mixed language code
- Submissions at head – One branch – All from source
- 20+ code changes/minute – 50% code change/month
- 5,500+ submissions/day – 120 million tests per day
- 80,000 builds per day – 20 million builds per year
- Auto code inspections – For low defect density
- 10X programming productivity improvement
- $150 million in annual labor savings (ROI as a result)

Micco, J. (2013). Continuous integration at google scale. Eclipse Con, Boston, MA.
Conclusion

- Agility is the **evolution of management thought**
- Confluence of traditional and non-traditional ideas
- Improves performance by over an order of magnitude

### Agile methods are ...
- Systems development approaches
- New product development approaches
- Expertly designed to be fast and efficient
- Intentionally lean and free of waste (muda)
- Systematic highly-disciplined approaches
- Capable of producing high quality systems
- Right-sized, just-enough, and just-in-time tools
- Scalable to large, complex mission-critical systems
- Designed to maximize business value for customers

“The world of traditional methods belongs to yesterday”
“Don’t waste your time using traditional methods on 21st century projects”

Books on ROI of SW Methods

- Guides to software methods for business leaders
- Communicates the business value of IT approaches
- Rosetta stones to unlocking ROI of software methods

http://davidfrico.com/agile-book.htm (Description)
http://davidfrico.com/roi-book.htm (Description)