# The Business Case for Agile Methods/Extreme Programming (XP)

Donald J. Reifer

Reifer Consultants, Inc.

d.reifer@ieee.org

Presentation at

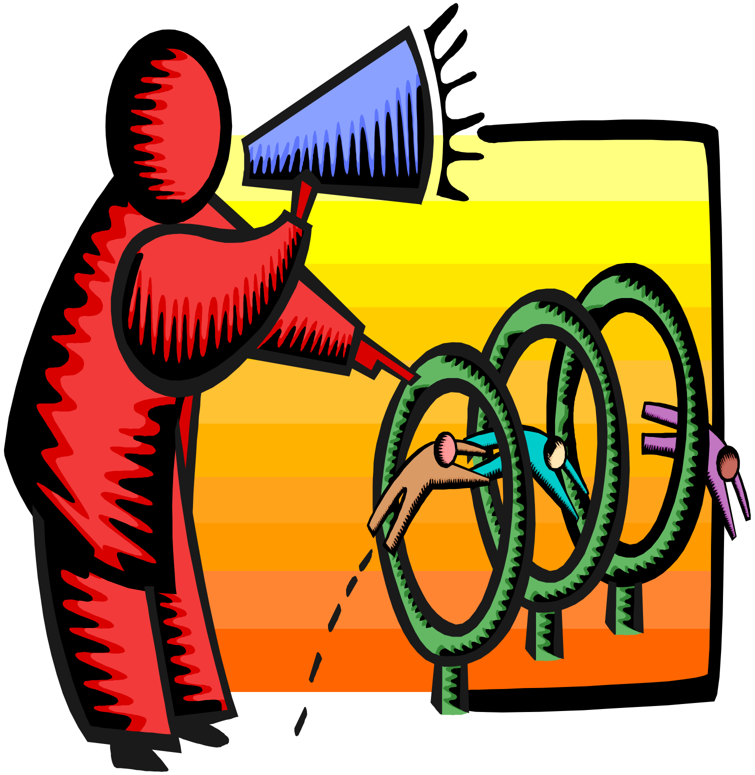
7th Annual PSM Users Group Conference

# Agenda

- Summarize results of study of 18 firms who used agile methods/XP on 78 projects
- Identify the key issues and important lessons learned
- Provide scorecard for agile method/XP performance in 12 application domains
- Using these data, postulate when and where it makes sense to use agile methods/ extreme programming

# The Agile Manifesto

1. **Individuals and Interactions** over **Processes and Tools**
2. **Working software** over **Comprehensive documentation**
3. **Customer collaboration** over **Contract negotiation**
4. **Responding to change** over **Following a plan**



**Value Maneuverability**

# Popular Agile Methods

- Crystal Methods
- Dynamic Systems Development Method
- Extreme Programming
- Feature-Driven Development
- Lean Development
- RUP Light
- Scrum

> "Deliver quickly, change quickly, change often."
>
> *Jim Highsmith*

> "Much of agile is about 'programmer power.'"
>
> *Bob Glass*

> "Agile recognizes people as the source of success"
>
> *Alistair Cockburn*

# Example: The 12 Practices of XP

- Metaphor
- Release Planning
- Testing
- Pair Programming
- Refactoring
- Simple Design

- Collective Ownership
- Continuous Integration
- On-site Customer
- Small Releases
- 40-Hour Work Week
- Coding Standards

*XP rewards demonstrable results; frequent demos of product software as it evolves*

# Study Demographics

| | No. | No. | | Average | No. |
|---|---|---|---|---|---|
| **Industry** | **Firms** | **Projects** | **Language(s)** | **Size** | **Large Jobs** |
| Aerospace | 2 | 8 | C++/VC | 46 KSLOC | 2 |
| Automobile | 1 | 3 | SQL/VB/HTML | 25 KSLOC | |
| Computer | 1 | 6 | C++/VB/VC | 35 KSLOC | |
| Consultants | 2 | 8 | SQL/VB/Java/HTML | 28 KSLOC | 1 |
| E-Business | 3 | 19 | SQL/VB/VC/HTML | 32 KSLOC | |
| Financial | 1 | 5 | SQL/Java/HTML | 58 KSLOC | server apps. |
| Gas/Oil | 1 | 4 | C++/VB/HTML | 55 KSLOC | 1 |
| Manufacturing | 1 | 3 | SQL/VB/Java/HTML | 22 KSLOC | |
| Researchers | 2 | 5 | C#/C++/VC/VB | 42 KSLOC | 1 |
| Scientific | 1 | 2 | C++/VB/VC | 28 KSLOC | |
| Software | 1 | 7 | C#/C++/VC/HTML | 32 KSLOC | 2 |
| Telecom | 2 | 8 | C++/VC/HTML | 58 KSLOC | 3 |
| **Totals** | **18** | **78** | ***Large project is over 100KSLOC*** | | ***10*** |

# Characteristics of Agile Projects

- For the most part, agile methods projects could be characterized as:
  - Short: One year or less in duration (many shorter)
  - Risky
    - Viewed initial use of agile as an experiment
    - Took on jobs with high volatility afterwards
  - Staffed with the high performers
    - Motivated, experienced and committed troops
  - Applications were mostly precedented
    - Several major exceptions
  - Projects characterized by high degree of required development flexibility
  - Architectures were stable
    - Mostly client-server
  - High degree of team cohesion
  - Some skepticism, but mostly enthusiastic
  - Anti-process attitude

# Five Surprising Findings

1. Most firms outside of E-Business were rated SW-CMM level 2 or higher
2. Requirements were initially stable (for early projects)
3. Architectures were well-bounded
4. Workloads towards end of project increased due to refactoring workloads
5. Duration prediction adhered to a square rather than cube root relationship with effort

# Looking at the Business Case (Surrounding Change with Numbers)

- **Business Case** = the materials prepared for decision-makers to show that the proposed idea is a good one and that the numbers that surround it make sound financial sense

- For agile, based on the hard data
  - Deliver product with equivalent quality in half the time
  - Productivity and cost for projects about the same
- Based on the soft data
  - Ability to reduce project volatility and risk by delivering working versions of the product frequently

# Summary of Analysis Results

- **Hard Data**
  - **Productivity gains**
    **14 to 25%**
  - **Cost reduction**
    **7 to 12%**
  - **Time to market gain**
    **25 to 80% reduction**
  - **Quality improvement**
    **On par with past experiences**

**Note –** Hawthorne effect may apply due to small sample size (18 firms, 78 projects)

- **Soft Data**
  - Mostly anecdotal evidence
  - Many early adopters used opinion surveys to understand results on pilot projects
  - Most used intangibles to build b-case for XP/agile methods
  - Most argued passionately for continued use of XP/agile methods after pilot projects
  - Most pressed to work issues in scaling, metrics & tech transfer
    - Additional processes needed for scaling agile to large projects
    - Metrics and measures currently
    - used don't provide needed insight
    - Seem to be relearning tech transfer lessons learned especially when trying to bridge the chasm

# Scaling Issues
# (Results of 2003 Banff Workshop)

- How to scale agile for non-pure agile projects?
- Guidelines for non-sweet spot agile projects?
- How to augment agile to fit large projects?
- How to address legacy, COTS, components within agile projects?
- How to scale agile within an enterprise across applications?

- How to handle dispersed development within agile projects?
- Who does integration testing as projects get better?
- What is agile (when polluted)?
- What project management practices do we use?
- How do we respond to RFP's when embracing agile methods?

# Scaling Lessons Learned
## (More Results of the Banff Workshop)

1. Scaling of agile methods will continue to happen whether you like it or not
2. Visibility for large projects can be increased via frequent planned time-boxed releases
   - The shorter the cycles the better
3. Communications on large projects can be improved using daily meetings
4. When scaling large projects, you can use a combination of compatible agile and traditional (plan driven) methods
5. When scaling large projects, empower your business analysts to be the voice of the customer

# Metrics/Measurement Issues

## Common Issues

- Schedule & progress

**Focus on working code as deliverable**

- Resources & cost

**Focus on many and frequent code deliveries**

## Measurement Category

- Milestone performance
- Work unit progress
- Incremental capability
- Personnel
- Financial performance
- Environment & support resources

## Sensible Measures

- Rate of progress
- Iteration performance
- Increment content and functionality
- # of people
- Iteration performance
- Environment & support resources still make sense

# Metrics/Measurement Issues

### Common Issues

- Product size & stability

**Test first focus**

- Process performance

**Focus on product, not process**

### Measurement Category

- Physical size and stability
- Functional size and stability
- Process compliance
- Process efficiency
- Process effectiveness

### Sensible Measures

- # tests developed and exercised
- Refactoring rate
- # of user stories/ use cases supported
- Refactoring rate
- Not important
- Productivity
- Cycle time
- Time to market
- Rework

# Metrics/Measurement Issues

## Common Issues

- Customer satisfaction

**Customer works as member of develop. team**

- Worker satisfaction

**Life style focus**

## Measurement Category

- Customer feedback
- Customer support
- Worker feedback
- Worker support

## Sensible Measures

- Feedback during development, not after the fact
- Customer support natural fallout
- Happy programmers
- Low turnover
- High morale
- High productivity
- Fewer complaints

**New metrics and measures are needed for Agile/XP Projects**

# Technology Transfer Issues

- Startup time and costs seem to be higher than expected
  - Good books/articles on topic
  - Not a a lot of training courses
  - Experts too busy with others
- Need accepted definition of what agile methods mean
- Lots of ideas, few specifics on how to make agile methods part of existing processes/practices
  - All or nothing attitude by some proponents (religious arguments abound)

- Few tools available to help to mechanize methods
  - Rely on manual techniques
  - Tools that exist are expensive and are mostly environments for collaboration
- Biggest problem making the giant leap forward to use on other projects

***Could have put this chart up for any other set of Methods as they were being transitioned to use***

# More Barriers to Adoption
## (Results of 2003 USC Workshop)

- PM/PEO credibility
- Customer credibility
- Paradigm change
  - Contracting
  - Organizational roles and responsibilities
  - PMR/earned value
- New metrics
- New skills

- Perceptions/ misperceptions
- Technical/transition infrastructure
- Agile standardization/ consolidation
- CMM/CMMI compatibility
- ROI/business case

# Balanced Scorecard/Industry

| | Projects | Budget | Schedule | Quality |
|---|---|---|---|---|
| **Industry** | **Complete** | **Perform** | **Perform** | **Perform** |
| Aerospace | 8 | High | High | Below |
| Automobile | 3 | Avg. | High | Par |
| Computer | 6 | Avg. | High | Par |
| Consultants | 8 | High | High | Above |
| E-Business | 19 | High | High | Above |
| Financial | 5 | High | High | Par |
| **Totals** | **49** | **Ratings indicate XP shows promise** | | |

**Internet projects dominate as large firms put much of their infrastructure support (travel, administration, enrollment, etc.) on the web**

# Balanced Scorecard/Industry

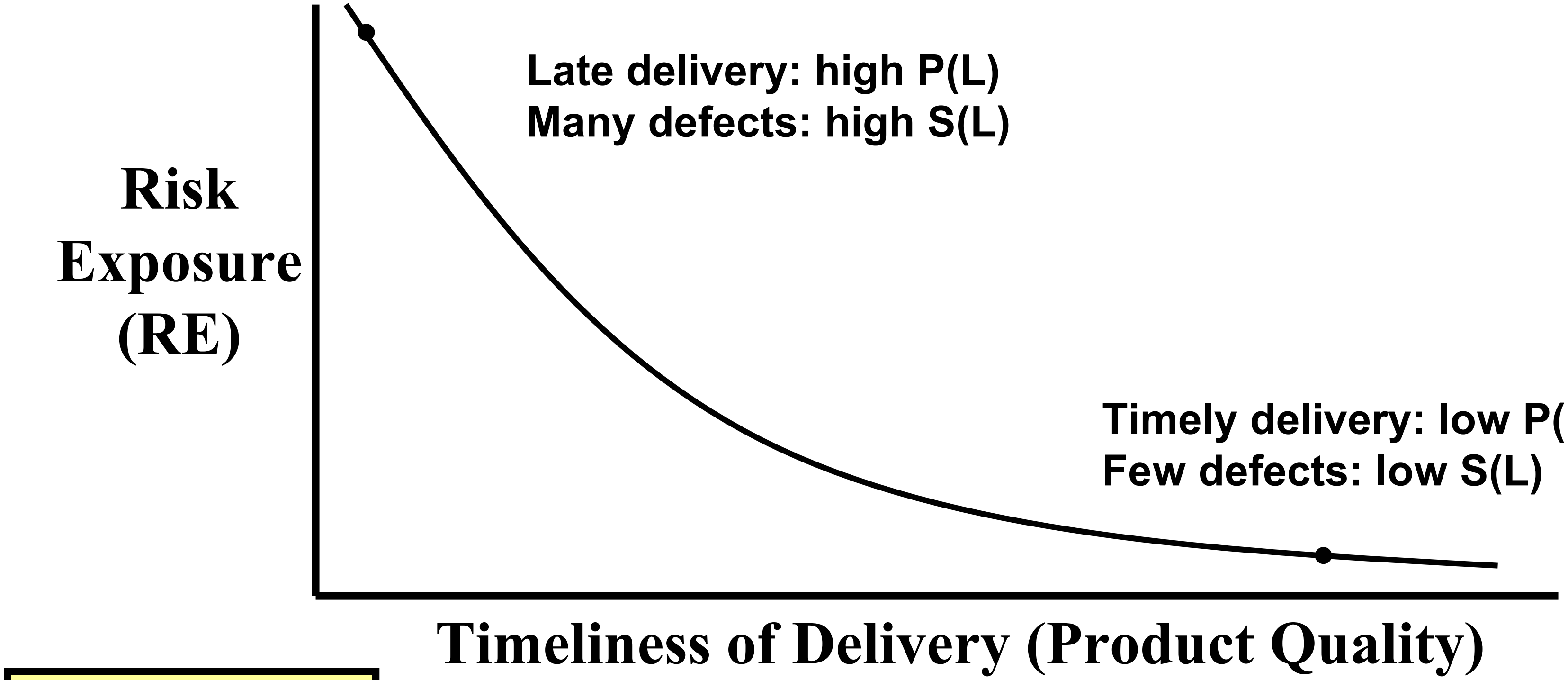
| Industry | Projects Complete | Budget Perform | Schedule Perform | Quality Perform |
|---|---|---|---|---|
| Gas/Oil | 4 | High | High | Par |
| Manufacturing | 3 | High | High | Par |
| Researchers | 5 | Avg. | Avg. | Par |
| Scientific | 2 | Avg. | Avg. | Below |
| Software | 7 | Avg. | High | Par |
| Telecom | 8 | High | High | Par |
| Totals | 29 | | | |

**Ratings indicate XP shows promise**

**Those estimating/managing budgets/schedules for agile projects/XP employ different practices than those used in traditional organizations**

# Quantifying Risk Exposure (RE) via a Profile: Timely Delivery

- Loss due to late delivery of products to market



Late delivery: high P(L)
Many defects: high S(L)

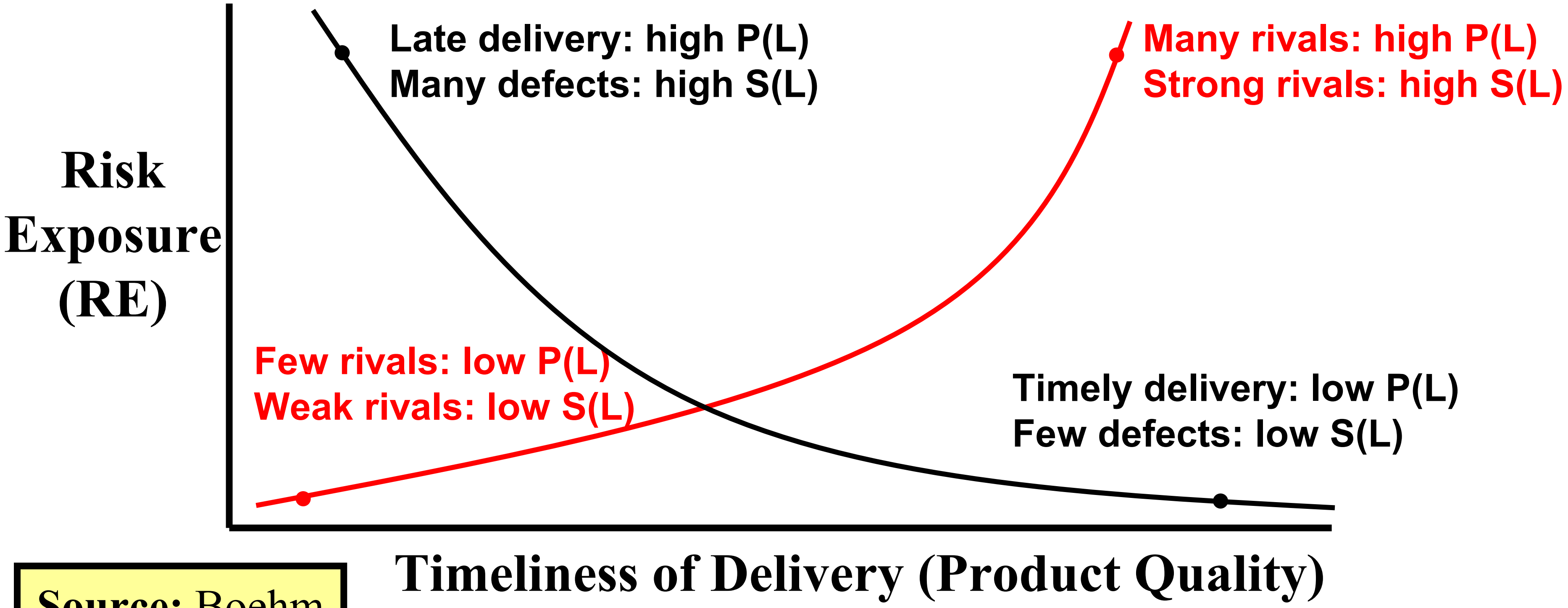
Timely delivery: low P(L)
Few defects: low S(L)

**Source:** Boehm

# Example RE Profile: Timely Delivery

- Loss due to late delivery of products to market

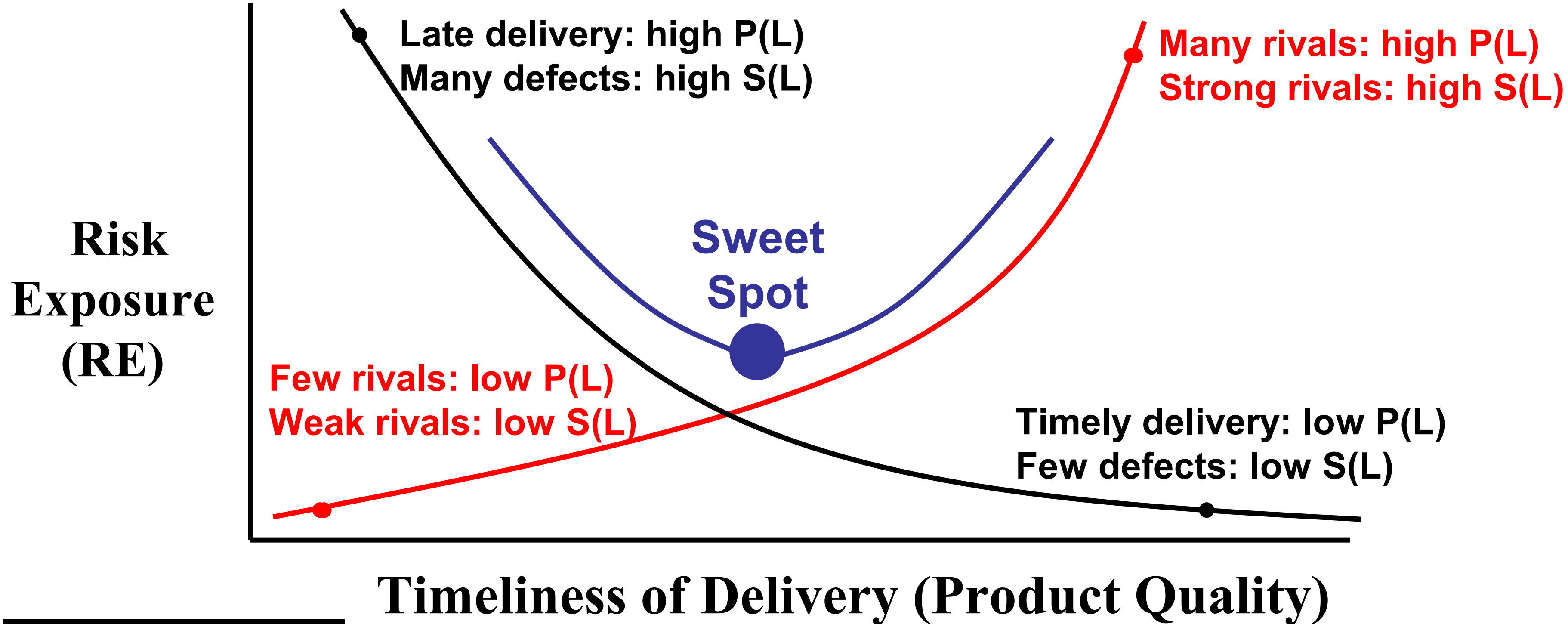
- Loss due to market share erosion



**Source:** Boehm

# Example RE Profile: Timely Delivery

## - Sum of Risk Exposures



**Source:** Boehm

# Eight Critical Success Factors

1. Time-to-Market the Focus
2. Proper Application Domain
3. Applicable Project Size
4. Volatile Requirements
5. Stable Architecture
6. Done by Skilled, In-House Team
7. Committed Customer
8. Suitable State of Organizational Readiness

# Some Additional Thoughts

- One firm argued that agile methods and PSP/TSP were compatible when integrated
- Another organization focused on weekly deliveries to the customer as testimony of the advantages of the methodology
- A third organization stated that what they were observing was a return to the chaos of the 1960's
- What do you think?

# Some Recommendations

- Clearly understand what is meant by agile methods
  - Variants/invariants
- Fit methods properly
  - Use lessons learned
- Focus on capturing more “hard” data
  - Use to convince the skeptics/ease the transition

- Introduce methods slowly and carefully
  - Address resistance to change
  - Provide startup guides and “how to” checklists
- Try to make the methods you adopt part of your process
- Do what makes business sense

# Would You Use Agile Methods/XP?

- **Would You Use?**
  - Yes, am using
- **Who is You?**
  - RCI, Cohesion Force and Raytheon
  - Test bed in Huntsville at customer site
- **What Products?**
  - Tools that automate protection technology
  - Demonstration scripts and conduct

- **Under What Conditions?**
  - Small tech demo project
    - 3 people, 20 KSLOC
  - Requirements volatile
  - Architecture of tools fluid
  - Focus is supporting successful demonstration
- **Why?**
  - Need to iterate based on experimental results
  - Need to show the customer progress

# Conclusions

- Lots of hype out there, some supported by fact
- Data shows agile methods have promise
- Need to learn more and understand how to make a leap forward
- Need to focus more on how to scale and transfer the technology

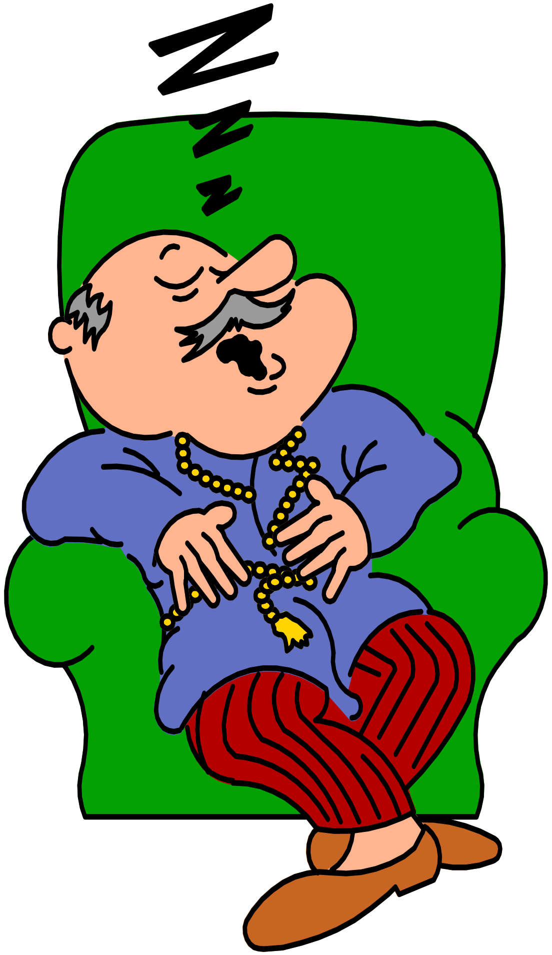
# Final Remarks

- "Technology travels with people. You can't just throw it over the wall and, because it is a a good idea, expect people to pick it up and run with it."

  *Chuck Geschke, Co-Founder of Adobe Systems*

- "He who does nothing, does nothing wrong"

  *Motto of the bureaucracy*



# Backup Slides

# What are Agile Methods?

- **Invariants**
  1. Process is cyclical with builds/increments done in parallel
  2. Organization is collaborative with participation by all stakeholders during the development
  3. Methods involved are considerably less formal than the traditional ones (less documentation)

- **Variants**
  1. The actual form of process used (spiral, incremental, etc.)
  2. Who the stakeholders are and the depth of their involvement
  3. Actual practices used under the banner of agile methods
  4. How informal the process is – degree of development flexibility