



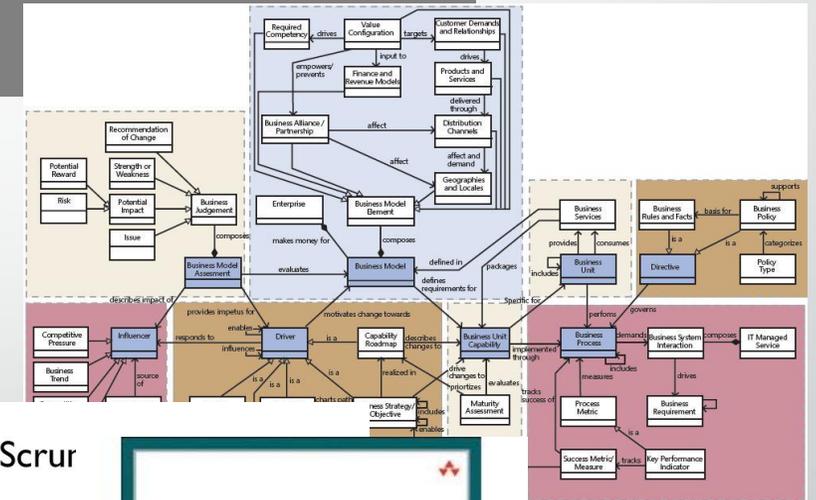
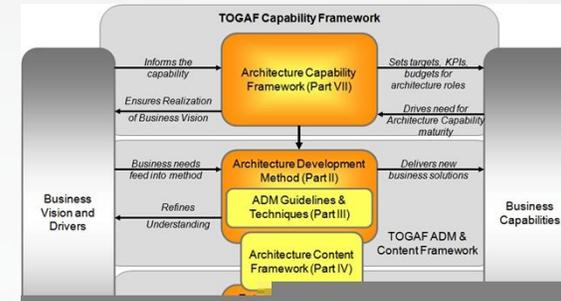
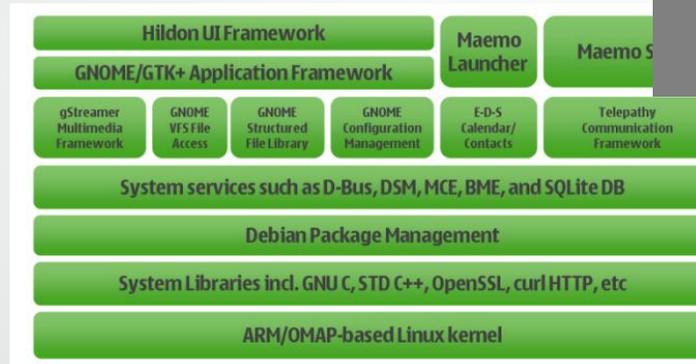
Delivering Open Architecture Solutions

By Applying Agile Principles

Presented to the IDGA
Interoperable Open
Architecture Conference
September 10, 2013
Martin J. Brown Jr.

Things I'm not talking about

- No representative architectures



- No discussion of specific agile methods

A collage of agile-related content, including:

- Scrum logo
- Book cover: **Extreme Programming Explained** by Kent Beck with Cynthia Andres, Second Edition. Slogan: **EMBRACE CHANGE**.
- Agile process diagram showing iterations (e.g., 4, 5, 6) and phases (e.g., 1, 2, 3, 4).
- Other agile-related diagrams and text.

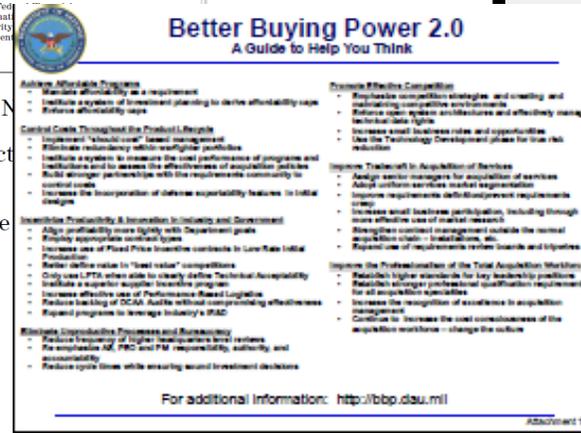
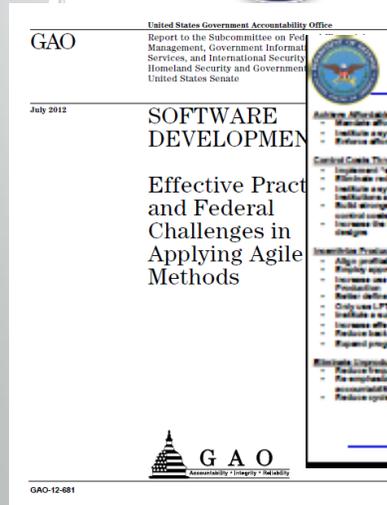
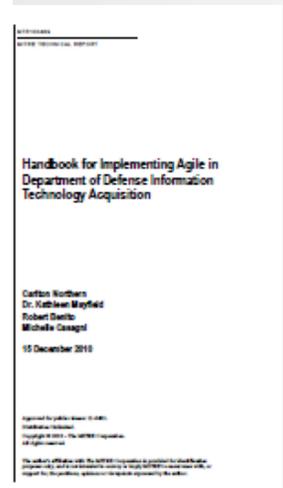
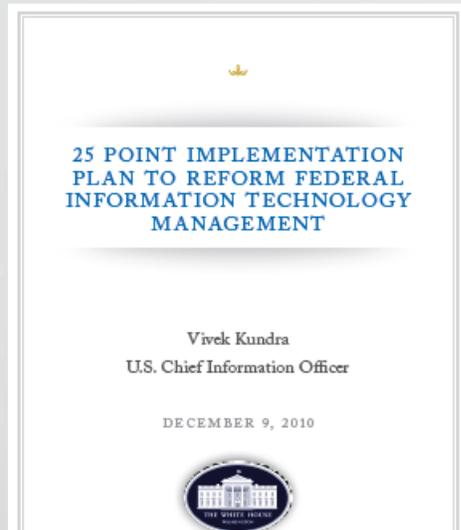
What I am going to talk about

- Why is implementing an Open Architecture so hard?
- Program manager's focus on improving effectiveness and efficiency
 - Open Architectures, agile principles and Better Buying Power
- Agile Principles applied to Open Architecture Implementation
 - Understanding the customer and delivering value
 - Simplicity, Last responsible Moment, Affordability
 - The role of the government manager

Open Architectures (Interoperable and Others)

- Interoperable Open Architectures are like the Holy Grail
 - A quest for the past 20 years
 - Isolated pockets of success amid large seas of disappointment
 - Navy's 2012 Open Architecture Strategy
 - “The current Naval Enterprise acquisition model is centered on highly integrated platforms with systems that are largely vendor locked, and expensive to acquire and upgrade”
- Continue to focus on the “What” instead of the “How”
 - Current Practice - Jump from implementation to implementation
 - Future Practice - Focus on how we implement open architecture solutions

Improvement is Needed



- Recognize the need to Improve Acquisition
 - Stress Affordability
 - Cost effective solutions
 - Encourage Innovation and Competition
 - Grow the Acquisition Workforce
- Acquisition Focus but not just an Acquisition Issue
 - Requirements changes
 - Budget changes
 - Product vs. Program view

BBP, Open Architectures and Agile

Better Buying Power	Open Architectures	Agile Concepts
Achieve Affordable Programs & Control Costs throughout lifecycle	✓	✓
Incentivize Productivity and Innovation	✓	✓
Eliminate Unproductive Processes and Bureaucracy	✓	✓
Promote Effective Competition	✓	
Improve Acquisition Workforce		✓

Agile Methods Have Value in the Government Environment

- SEI Considerations for Using Agile in DoD Acquisition – 2010*
 - Define Agile as: “ An iterative and incremental (evolutionary) approach to software development that is performed in a highly collaborative manner by self-organizing teams within a governance framework with “just enough” ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.”
- GAO Effective Practices and Federal Challenges in Applying Agile Methods – 2012
 - Identified 32 practices and processes effective for the application of agile to IT projects and 14 challenges to the implementation of agile in the Federal Environment

True Power of Agile Often Missed due to a Focus on Speed

- Both SEI and GAO focused on Short Iterations as key elements in adopting agile.
 - Speed of delivering working software does improve Return on Investment
- Both refer to the Agile Mindset as some mythical, unattainable state
 - Agile Mindset maximizes value to the customer
 - Develops partnerships with contractors, customers and stakeholders
 - Enhances transparency to create a shared solution space
 - Iteration Length is a tool used to deliver value
 - Understand that time and budget are constraints

Agile Program Manager

"I can't deliver everything you want, but I can deliver the most valuable things you want."

"You need to work with me to ensure that my efforts remain aligned with your priorities."



Working with customers in a transparent environment

Developing an Open Architecture

Current Approach



Need for efficiency



Senior Manager
establishes a
program



Group of really smart people
go off to define and design the
“more perfect architecture”



Which are ignored or
waived by most
programs citing a lack of
funding to implement

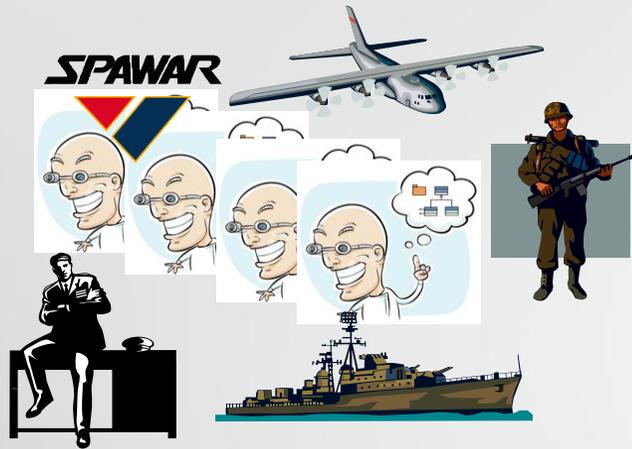


Finally resulting in the
perfect set of open
standards



Analysis produces a mountain
of documents citing the need
for more study

An Agile Approach



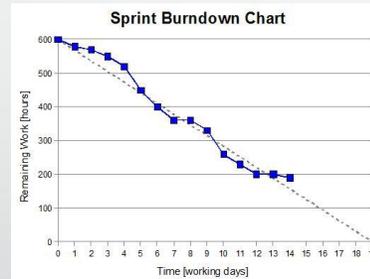
Assemble key stakeholders focused on current limitations appropriate for enterprise solution



Evolve and Grow the Architecture



Jointly develop a program investment vision, identify key features, establish initial priorities, define done



Implement and deliver working solution



Develop initial set of user stories. Provide initial estimate of size.

Oct	Nov		Dec
W12	W14	W16	W18
..
..
..
..

Implementation Schedule

Develop initial release plan

What changed when we applied agile principles?

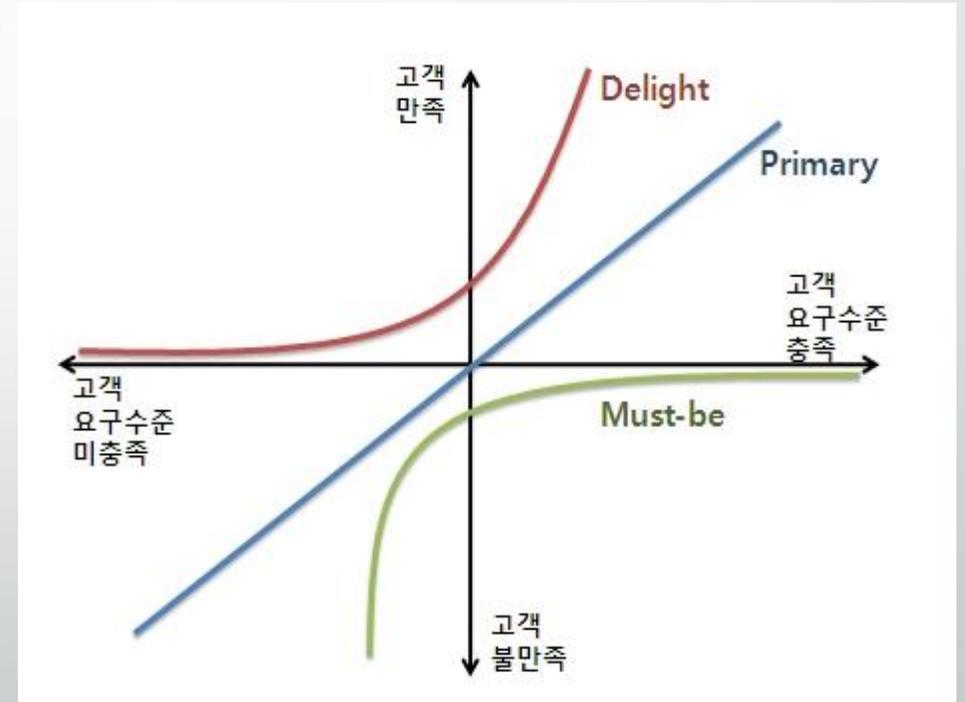
- Customer Determines Value
 - Defined in the customer's terms
- Vision Statement
 - Defined using broad objectives rather than specific requirements
- Limited up front design with a plan to revisit and elaborate
 - Execution is a bottom up process
- Development priorities set by customers and stakeholders
 - Dynamic process
- Management complexity increases

Customer Focus

- Know who your customer is
 - Potential problem with Open Architecture Solutions
 - Often the focus is on the warfighter
 - Real customers are often the application developers
- Listen to the customer and deliver what they want
- Progressive elaboration to decompose requirements just in time
- User Stories using INVEST framework
 - Independent
 - Negotiable
 - Valuable
 - Estimable
 - Small
 - Testable

Understanding What is Important to the Customer

- Kano Analysis – Four basic classifications
 - Must Haves – Baseline Features, Dissatisfaction when absent
 - Satisfiers – value added, the more the better
 - Delighters – exciting but unexpected features
 - Indifferent – customers don't find value in these



The Agile Vision Statement

- Characteristics
 - Short, concise, easy to understand
 - Defined at the Feature Level rather than specifications
 - Establishes the Program Boundaries
 - Includes a definition of done
- Tips for Creating
 - Design the Box, Write the review
- Should flow to any contracts

Managing the Agile Effort Requires Change

- Agile is based on cooperation and collaboration
 - Partnerships with contractors, customers, stakeholders
 - Working with self-directed teams
- Changes in roles and responsibilities
 - Program Managers become “Product Owners”
 - Increased transparency
- Manage to decision points (Last Responsible Moment)
 - Understand what you need to know and when you need to know it
 - Approach to risk
 - Traditional governance often gets in the way

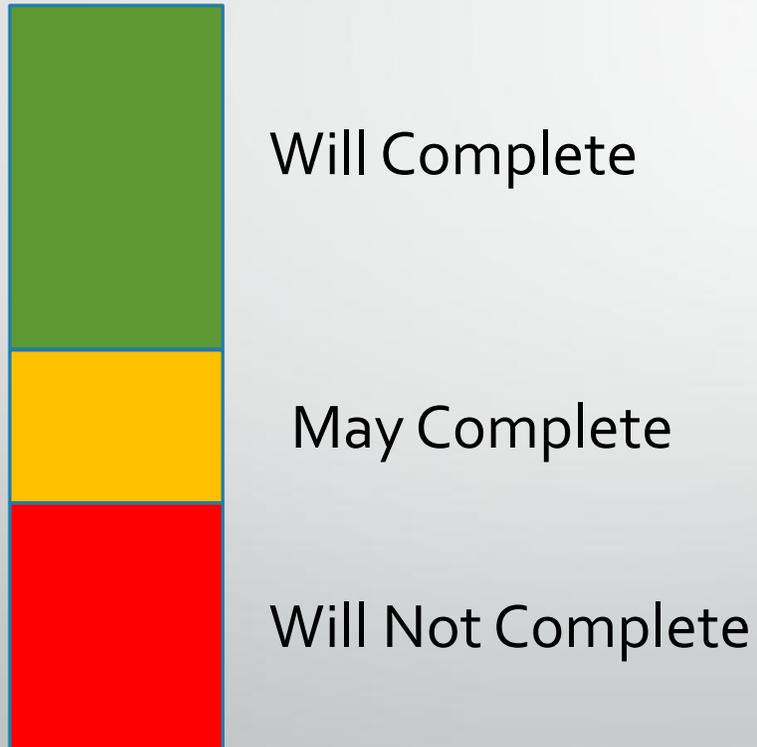
Progressive Elaboration, Simplicity, and Last Responsible Moment

- Progressive Elaboration
 - Guides how agile programs evolve over time
- Simplicity
 - The process of maximizing the value of work not performed
 - Relates directly to BBP objective of eliminating unnecessary processes and bureaucracy
 - Appropriate level of planning at each point in the program
- Last Responsible Moment
 - Key agile concept
 - Decisions need to be made at the point that a valuable alternative will be lost

Transparency

- Establish realistic expectations

- Estimated story size
- Assumed velocity



- Updated as new information is available

- Estimated story size
- Actual velocity

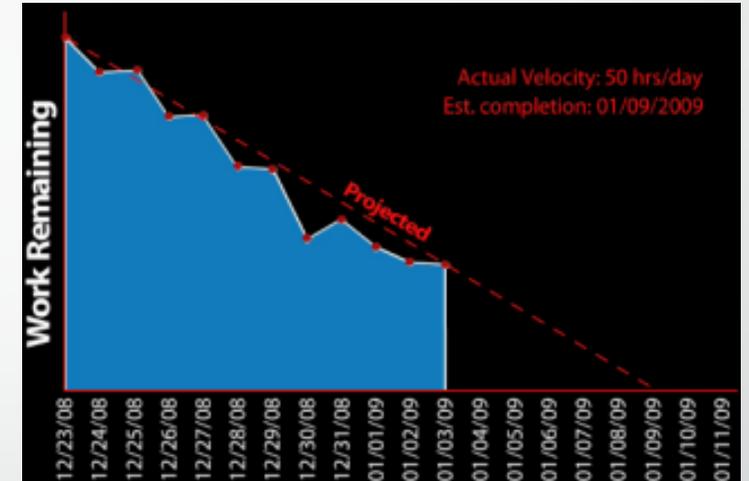


Managing in an Agile Environment

- Dynamically Managing Three Elements
 - Product development
 - Making sure the team is focused on top customer priorities
 - Building what is asked for and nothing more
 - Product Knowledge
 - Identifying and answering key questions about the product
 - Risk reduction efforts
 - Generally implemented as spike stories
 - Process Knowledge
 - Gaining understanding and improving how we are developing
 - Retrospectives

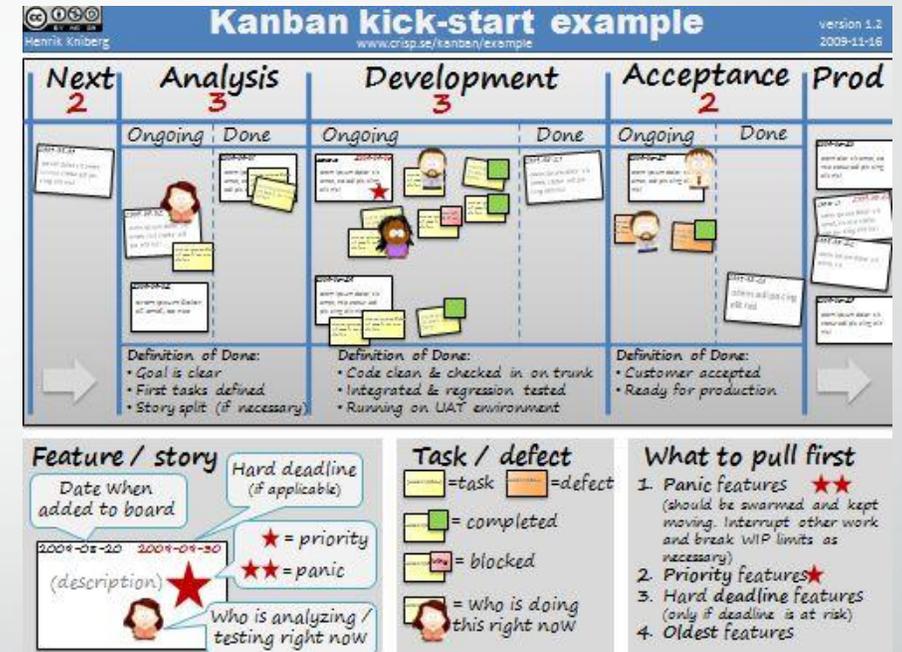
Agile Approach to Risk

- Agile Philosophy is to attack risk
 - Fail Fast
- Unresolved Risk increases in severity as project progresses
- Actively manage risk resolution through spikes
 - Stories/tasks designed to answer key risk questions
 - Time boxed to manage investment in answering risk issues
- Supports decisions at the Last Responsible Moment



Managing the Implementation

- Limit Work in Progress
 - Key element in Kanban and Lean
 - Cannot start a new task until one is done
- Focus on completing tasks
 - No value until the task is done
 - Avoids the 95% complete “and holding” situation
- Dispels multi-tasking myths
 - Humans actually perform task switching
 - 2 – 3 simultaneous tasks allows use of dead time
 - Beyond three we lose efficiency during reboots



Applying Agile Concepts to Open Architecture Development

- Customer defined value
 - Identify and work with customers throughout the lifecycle
 - User Stories place requirements in context of the customer
 - Close collaboration throughout the lifecycle
- Technical excellence and attention to detail
 - Allow the requirements to evolve
 - Working Architectures emerge from the effort
- Product vs. Program View
 - Investments aligned with IT Strategy
 - Partnerships instead mandates

Conclusion

- We (the Government) need to focus on how we implement open architectures
- Good solid architectures, based on open standards, can emerge from the bottom up
- Resource allocation needs to evolve to an investment focus rather than a program focus
- Agile incorporates transparency to allow stakeholders to understand how the program is executing
- Managing an Agile effort is different - roles and responsibilities change