Tailoring the SE Process to Effectively Complement the SW Agile Development Process

National Defense Industrial Association (NDIA)
12th Annual Systems Engineering Conference
October 26-29,2009

System Engineering Effectiveness Track,
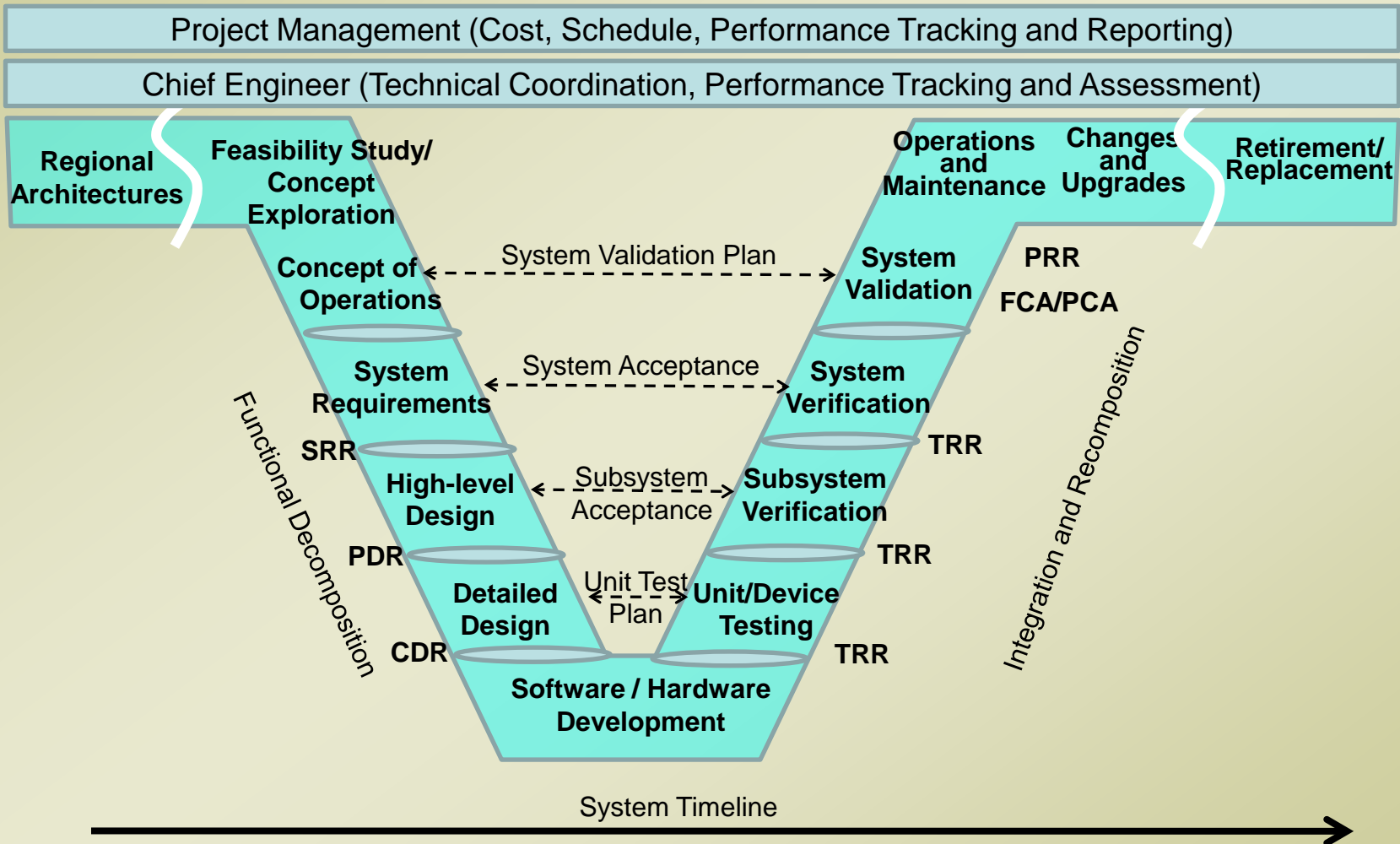**Wednesday October 28, 2009**

William Lyders, ASSETT Inc.

Outline

- **Majority of Programs today use variations of the Traditional Waterfall Model**

- **An Iterative/Agile Model approach provides some important benefits not easily possible with a single Waterfall Model**
  - ➢ **End User Assessment**
  - ➢ **Accommodate Requirement Changes**
  - ➢ **Early delivery of production features**

- **System Engineering process was tailored to complement a Software Agile Development process**

- **Each of the Phases in the Agile Cycle is shown to incorporate traditional activities while providing the desired benefits in the Agile process for both SE and SW processes.**

What is Systems Engineering?

- **Systems Engineering is:**
  A <u>rigorous discipline</u> for successful <u>system development and integration</u>. It <u>elaborates business needs</u> into traceable and testable requirements, <u>establishes system baselines</u> and <u>integrates and delivers</u> the full system solution.

- **Systems Engineering strengthens the development process by adding:**
  - A more <u>rigorous approach </u>to requirements definition, management, baseline management, and traceability through delivery,
  - A series of <u>formal reviews </u>linked to key lifecycle events,
  - Formal linkage to the QA process throughout the project lifecycle; accelerating the detection and resolution of defects earlier where it costs less to resolve,
  - <u>Verification</u> that the Technical Solution meets the <u>requirements</u> and <u>Validation </u>that it meets the customer needs.

- **It is also:**
  - A key enabler to achieving higher CMMI maturity levels
  - Industry-recognized as a core capability required for <u>complex systems integration</u> programs.

Traditional Systems Engineering "Vee" Model

Project Management (Cost, Schedule, Performance Tracking and Reporting)

Chief Engineer (Technical Coordination, Performance Tracking and Assessment)

Regional Architectures

Feasibility Study/ Concept Exploration

Concept of Operations

System Requirements

SRR

High-level Design

PDR

Detailed Design

CDR

Software / Hardware Development

Functional Decomposition

System Validation Plan

System Acceptance

Subsystem Acceptance

Unit Test Plan

Operations and Maintenance

Changes and Upgrades

Retirement/ Replacement

System Validation

PRR

FCA/PCA

System Verification

TRR

Subsystem Verification

TRR

Unit/Device Testing

TRR

Integration and Recomposition

System Timeline

Agile Model - Areas of Innovation

Innovation: Agile Development Process

Project Management (Cost, Schedule, Performance Tracking and Reporting)

Chief Engineer (Technical Coordination, Performance Tracking and Assessment)

Feature Set #1
Feature Set #2
Feature Set #3
Feature Set #4
Feature Set #5
Feature Set #6

Chief Engineer Provides Coordination

Chief Engineer Provides Coordination

SRR

PDR

CDR

1 Month Test Cycle

1 Month Development

System Timeline

Functionally Decompose System Capabilities into Feature Sets that Have Operational Significance
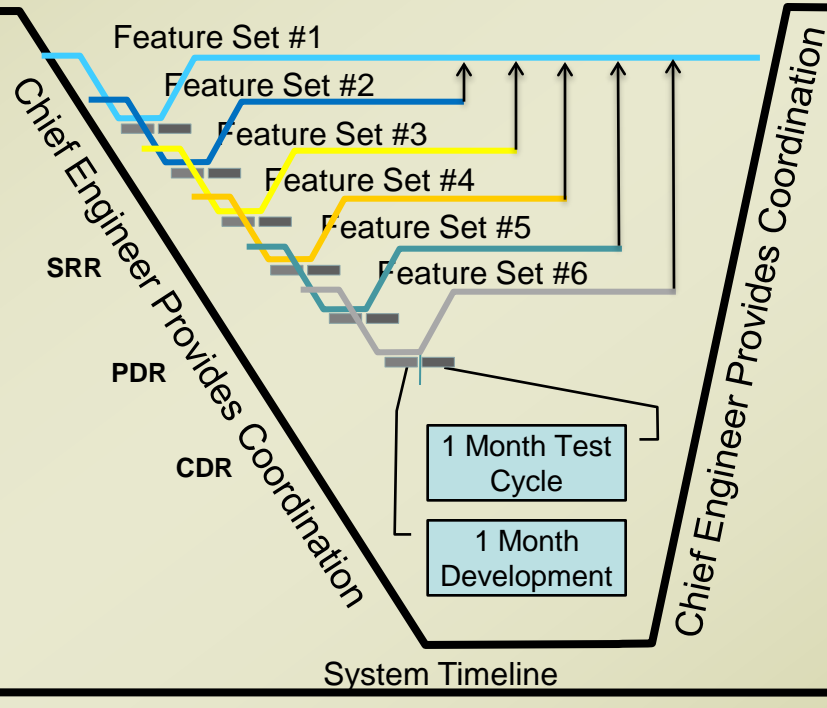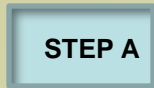
Benefit: The Slope for the Integration & Testing Phase Is Much Steeper Since Features Have Already Been SE Tested as a Part of The Development Process

Feature Sets Provide an Operational Capability that Can Be Tested and Validated Early By the Client

Transitioning from Waterfall Model to an Agile Model

ASSETT

Plan/ Conceptualize

STEP A

Requirements/ Analysis

STEP B

Design

STEP C

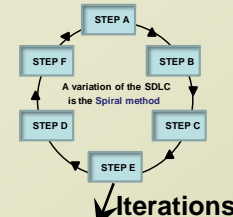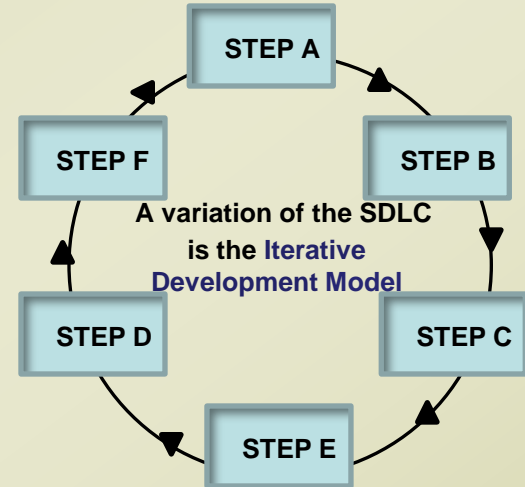**Build**

STEP D

Test (V&V)

STEP E

Implement/ Maintain/Train

STEP F

The Traditional SE Process is based on the Waterfall Model and operates in a single cycle of all the requirements

The Iterative Development Model has one major differentiation from SDLC: operates on only a limited set of requirements each iterative cycle and creates limited results very quickly.

STEP A
STEP F
STEP B
STEP D
STEP C
STEP E

A variation of the SDLC is the Iterative Development Model

STEP A
STEP F
STEP B
STEP D
STEP C
STEP E

A variation of the SDLC is the Spiral method

Iterations

Full system implementation

There are many versions of the Systems Development Life Cycle (SDLC) process.

A successful and widely used method over the years is the Waterfall Model.
• A logical & sequential set of process steps,
• Criticisms:
  ➢ Considered a "rigid" & "inflexible" procedure
  ➢ Requirements usually uncertain at the beginning
  ➢ Long process with no working version until late in the process

Multiple iterations of the Iterative Development Model are required to complete a full system implementation.
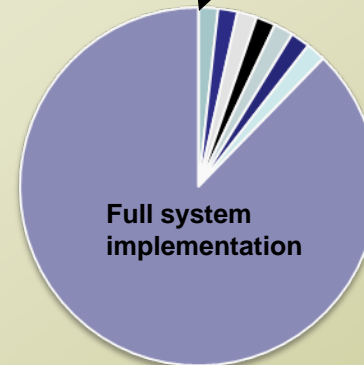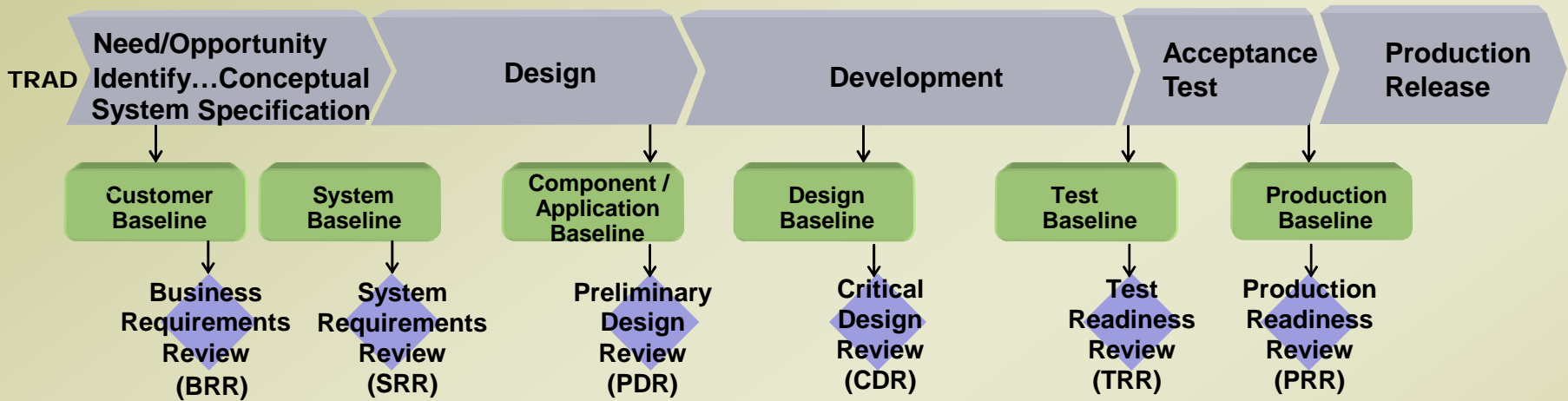
One variant of the Iterative Development Model is the Spiral Model typically used for large systems

Another variant, sometimes used for smaller projects, is the Agile Model for software development

6

System Agile/Traditional Engineering Process Overview

ASSETT

TRAD	Need/Opportunity Identify…Conceptual System Specification	Design	Development	Acceptance Test	Production Release

Customer Baseline → Business Requirements Review (BRR)

System Baseline → System Requirements Review (SRR)

Component / Application Baseline → Preliminary Design Review (PDR)

Design Baseline → Critical Design Review (CDR)

Test Baseline → Test Readiness Review (TRR)

Production Baseline → Production Readiness Review (PRR)

Traditional DoD Projects have specific baselines and review milestones

← - - - - - - - - - - - - - - - - Iterative Cycle - →

← - - - - Phase - - - - →   - - - - →   ← - - - - Iteration

SRR Package   PDR Package   CDR Package   System HW & SW   Production Support

AGILE	Inception	Elaboration		Implementation			Transition	Maintenance
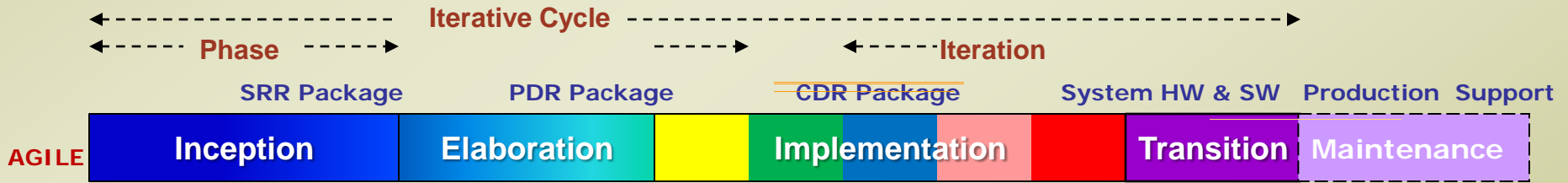
The ASSETT Agile Process maintains those specific baselines and review milestones
…but is implemented with new phase nomenclature & multiple iterations of operational capabilities

User Needs & Technology Opportunities

Milestones: △A △B △C

ACQ FWK	Concept Refinement	Technology Development	System Development & Demonstration	Production & Deployment

7
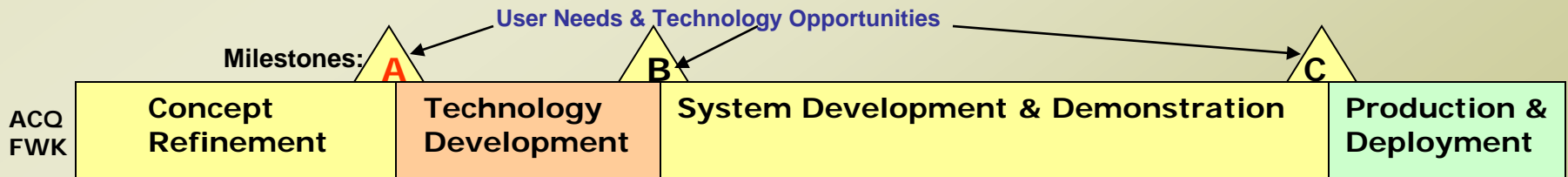
Unified System Development Process

Inception	Elaboration	Construction	Transition

iterative

Systems

- Domain Engineering
- Operational Concepts
- System Architecture
- System Requirements/Use Cases

- System HLD/PDR
- Key Systems Proofs-of-Concept
- Requirements to Features

- Development Support/Requirements Integrity
- Test Plans/Procedures by Feature
- Independent Iteration/Regression/Acceptance Testing Transition to Next Spiral Cycle

- System Requirements to Software Requirements

Software

- Understand Problem Domain
- Early Software Proofs-of-Concept

- Overall Software Architecture and High Level Design

- Design by Feature Build by Feature

- Transition to Demo Transition to Next Spiral

Based on the Unified Software Development Process
A good overview of UP can be found at http://en.wikipedia.org/wiki/Unified_Process

- **Inception Phase**
  - ➢ **Planning** – important risks identified and project roughly estimated
  - ➢ **Requirements Definition** – capturing/understanding customer requirements and generating simplified use cases for operational architecture
  - ➢ **Concepts** - good ideas developed into product vision, tentative architecture,
  - ➢ **Prototypes** – obtaining and trying new infrastructure and development techniques for implementation later
- **Perform Inception Phase for each Agile Cycle**
  - ➢ **1st cycle may have a long Phase**
  - ➢ **Later cycles may be very short**
- **Define initial requirements with some level of knowledge of complexity**
  - ➢ **In Agile if the complexity of a requirement/feature exceeds the Implementation plans later, then Agile allows flexibility to defer a feature to a later iteration or even a later cycle [Spiral Model does not allow this flexibility]**
  - ➢ **Requirement/Feature baselines under CM control…but iteration planning allows for changes to requirements and implemented features**

- **Elaboration Phase**
  - ➤ **Implementation Planning** – (Business architecture) Plan development by feature
    - Can defer features at customer request or move future features forward
    - Can establish full implementation iterations and "buffer" iterations

  - ➤ **Requirements Analysis/Feature Definition** – Create architecture model items
    - (Functional architecture) build feature lists for functional model based on analysis of requirements including detailed feature/iteration mapping
    - (Operational Architecture) creating detailed use case model

  - ➤ **Concepts and Prototypes** – (Physical architecture) refine concepts and infrastructure prototypes and hardware designs
    - Technology and operational realism tradeoffs
    - For SOA, map features into Services

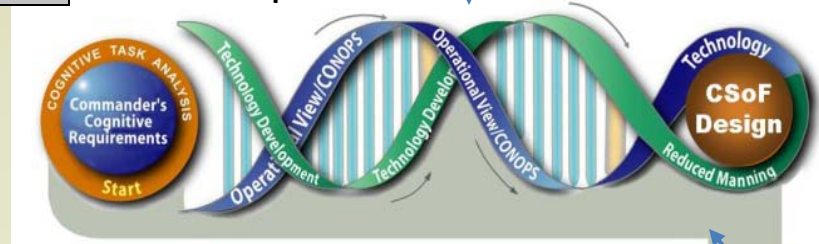 - ➤ **Architecture Definition** – Document above items in full System architecture

Missions: (Representative mission phases from the CSoF Operational Scenarios: Port Egress, Submerged Transit, and Intelligence Surveillance and Reconnaissance (ISR))
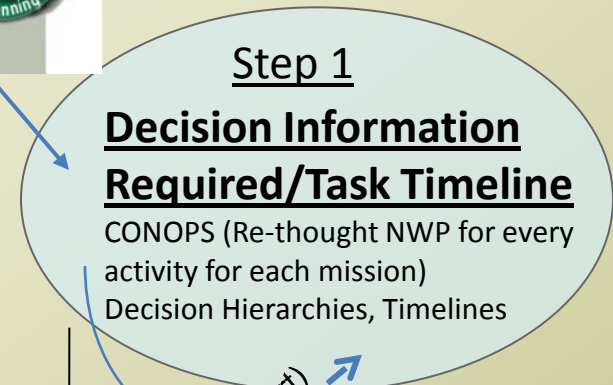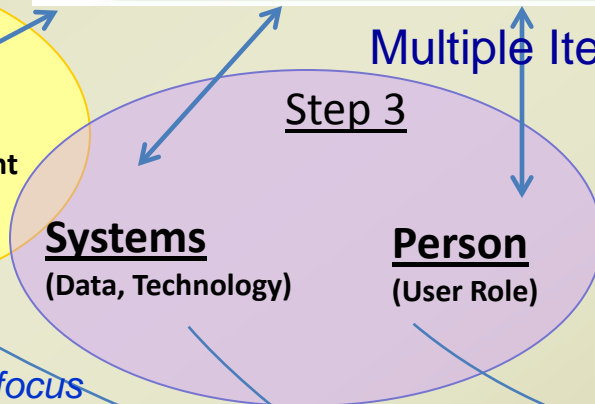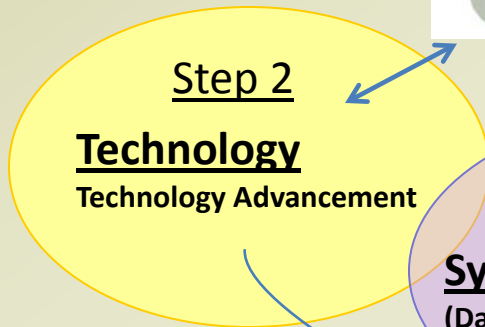
Decisions: To support Missions

Objective: Formulate data to define operational decisions & capture technology for TDS & ICD inputs

Double Helix process

Activities/Tasks

Operational View [DODAF]

Identifies What Needs to Be Done and Who Does It

Operational Elements

Information Flow

- wDx
- SME Interviews

Step 2

Technology
Technology Advancement

Multiple Iterations

Step 3

Systems
(Data, Technology)

Person
(User Role)

Step 1

Decision Information Required/Task Timeline
CONOPS (Re-thought NWP for every activity for each mission) Decision Hierarchies, Timelines

Proceeding Ahead: Refocus Data Gathering to narrow search and expand data attributes for modeling

Operators

TIME (work effort)

SEQUENCE (parallel, etc.)
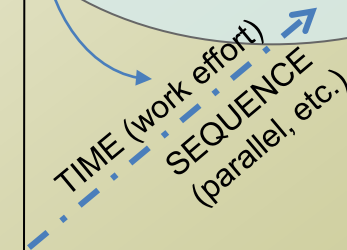
Resources

- **Implementation Phase** (Multiple Development Iterations within)

  - ➢ **During each implementation iteration period, perform each of the following activities below**

    - **Plan** – Revisit implementation status and re-plan feature development as necessary

    - **Requirement/Feature** – Revisit requirements/features for that iteration and <u>revise plans as necessary</u>

    - **Build** – develop all planned features for that iteration…Production code

    - **Test** – Conduct independent SE testing of software delivered from the last implementation iteration period. Conduct early <u>customer demonstrations</u> of new features – confirm requirements/features

  - ➢ **Implementation iterations are typically color coded or numbered so referencing functionality in planning and documentation is simplified.**

  - ➢ **The ability to <u>adapt implementation to complexity and changes in requirements</u> a key benefit in this Phase – customer feedback and development team productivity driven.**

12

- **Transition Phase**
  - ➢ **Conduct each of the following activities as necessary**
    - **Test (SE)** – Conduct <u>independent testing </u>of software delivered from the last implementation iteration period. Verify functional requirements/ uses cases for the developed software.
    - **System Integration/Regression Test** – Conduct the planned set of regression tests to verify that the new capabilities being implemented did <u>not affect any previously released capabilities</u>.
    - **Acceptance Test** – Conduct testing of the new capabilities with the customer witnessing the testing as an acceptance approach. Sometimes only a <u>demonstration </u>of the new capabilities needs to be done.
    - **Information Assurance (IA) Regression Test** – If the system requires IA certification, determine if and IA certification <u>re-certification testing </u>is required and perform any that is necessary.

  - ➢ **Formally release the new versions of the system, conduct training, and provide product assistance as necessary**

Summary

- **An Agile Model allows some special benefits over the traditional single cycle Waterfall Model**
  - ➤ **Requirements change flexibility**
  - ➤ **Observable features developed and requirements verified early**
  - ➤ **Early and often customer assessment of developed production features**
- **System Engineering process can be tailored to complement a Software Engineering Agile Development Process**
- **The SE-DE Agile process includes the traditional baselines, milestones, and provides early working products that is demonstrated to the customer for each cycle**

Q&A

Abstract

Tailoring the SE Process to Effectively Complement the SW Agile Development Process

Track: SE Effectiveness

At the 2008 NDIA SE Conference, the change to include Software Engineering topics was mentioned early in the conference. A key Software Engineering Process is the Agile (Iterative) Development Process, a version of the Spiral Process used by many companies. This presentation will identify how ASSETT Inc. has successfully tailored the traditional Systems Engineering waterfall process to complement its Agile Software Engineering Process. It will also show how we tailored the Test and Evaluation Process accordingly and incorporated operational concept designs, COOPEX events, and technology/ operational demonstrations early and throughout the tailored SE Process.

The Traditional SE Process is a popular version of the systems development life cycle model (SDLC) with a single Iteration: The waterfall [a.k.a Traditional or "Vee"] SE Process includes a single pass of the SE Process steps from requirements specification through design, build, and test & evaluations prior to delivery to the customer. Many customers are comfortable with this traditional acquisition process of work products but it does have its limitations.

The SW Agile Process Complements the Traditional SE Process Using Multiple Iterative Steps: A brief overview of the SW Agile [Spiral] Development Process will show that even though the comparable traditional SW Process steps were renamed and are performed multiple times during the system development life cycle, they are really the same types of activities and can be mapped within the traditional SE Process. The Agile Process results in demonstrable system building blocks at each iterative step. Also as each iterative step completes, key SE and SW Process decisions must be made, usually with customer input, and completed system features become available. This allows us to have multiple opportunities for customer interaction to jointly decide and prioritize upcoming iterations and operational/technology demonstrations that are important to the customer.

Tailoring the Traditional SE Process blends the best of both Processes: A tailored SE Process and a T&E Process have been aligned with the SW Agile Process and DoD Acquisition Process resulting in a very effective system development process. A description of the tailored SE, SW Engineering, and T&E Processes as they are performed in an iterative fashion are shown, including the work products and process step decisions. Another benefit of this tailored process is to allow for early operational realism, by conducting technical and operational demonstrations and CONOPS Exercises (COOPEX) at critical iteration completions derived from our Double Helix Methodology, Mission Driven Design (MDD), and Decision Centered (DCD) Design methodologies.

Author Biography

Tailoring the SE Process to Effectively Complement the SW Agile Development Process

Author Biography – Mr. Lyders is currently a Systems Engineering Manager and Lead Systems Engineer/ and Test Director for multiple projects at ASSETT, Inc. He has over 40 years of both systems engineering & project management experience in both federal software and commercial Information Technology (IT) development projects.

He has significant complex system specification, system information display design, system test and integration expertise, dockside, and at-sea testing experience developed through his federal work with multiple Sonar, Command and Control, and Submarine Combat Systems and multiple SBIR projects for the Navy. He is currently leading SE efforts for a new submarine Combat System of the Future (CSoF) at ASSETT for the Navy.

ASSETT is located at 11220 ASSETT Loop, Manassas, VA. Founded in January 2001, the company is focused on working with our customers to apply the systems engineering process to achieve quality deliverables that are affordable, and supportable, while meeting the demands for reduced time-to-market. ASSETT has 67 full-time and/or part-time employees ranging from recent Masters Degree graduates to experienced personnel with over 35 years of experience in the design, development, production, and support of large complex military systems.