SWX: The Software Extension to the PMBOK Guide® for Project Management

prepared and presented by

Richard E. (Dick) Fairley, PhD, CSDP

Software and Systems Engineering Associates (S2EA)

Agenda

- What is SWX?
- Why a software extension?
- Key elements of SWX
- The process
- Current status
- Lessons learned
- Questions?

What is SWX?

- SWX is an adaptation and extension of A Guide to the Project Management Body of Knowledge (*PMBOK® Guide*)
- SWX is a joint project of the IEEE Computer Society (CS) and the Project Management Institute (PMI)
  - a ten-member team; 5 from each organization
  - the first joint endeavor by each organization
  - copyright will be shared by the CS and PMI

SWX Team Members

IEEE CS Members	PMI Members
Dick Fairley, team chair	Dennis Stevens, co-chair
Ken Nidiffer	Jesse Fewell
Annette Reilly	Mike Griffith
Rich Turner	Cindy Shelton
Chuck Walrad	Krupakar Reddy

Why a Software Extension?

- The PMBOK® Guide is a generic document
  - intended as guidance for managing all kinds of projects
- More than 65% of the 400K+ members of PMI identify their work as IT or software related
- The mission of the CS is to develop products and services for the ~100K members
  - primarily through the Professional Activities Board
- Other PMBOK® extensions include:
  - the construction extension
  - the government extension

Why a Software Extension? (2)

- Software is an intangible product
  - which lends itself to a variety of development approaches
  - unlike physical products
- Software developers are knowledge workers
  - who develop innovative solutions to new problems
  - by sharing knowledge and outcomes in a closely coordinated learning environment

Why a Software Extension? (3)

- Traditional project managers will learn about methods, tools, and techniques for managing software projects
  - what's the same; what's different?
- Software project managers and practitioners will learn how traditional methods of project management can be tailored for software projects
  - what can we learn and apply?

Why a Software Extension? (4)

"managing a large computer programming project is like managing any other large undertaking—in more ways than most programmers believe. But in many ways it is different— in more ways than most professional managers expect"

The Mythical Man-Month, Anniversary Edition, Frederick P. Brooks, Jr., Addison Wesley, 1995; pp. x.

The PMBOK® Guide

- The PMBOK® Guide – Fifth Edition has just been released
- It includes 47 processes grouped into 10 knowledge areas
  - each process is described by
    - Inputs,
    - Tools and Techniques, and
    - Outputs
- The PMBOK® Guide is a guide to the BOK, not the BOK
  - it includes commonly accepted good practices
  - but does not attempt to cover every possible situation
- The PMBOK® Guide is about 500 pages in length

A Key Statement in the PMBOK® Guide

- From Section 2.4 of the PMBOK® Guide – Fifth Edition

  "Project life cycles can be described as falling somewhere in a continuum from predictive or *plan-driven* approaches at one end to adaptive or *change-driven* approaches at the other."
- As modified in SWX:

  "Software project life cycles can be described as falling somewhere in a continuum from highly predictive approaches at one end to highly adaptive approaches at the other."
- Note:
  - adaptive projects also have plans
  - predictive projects also have changes

SWX Figure 2-1

Highly Predictive **Predictive** **Adaptive** **Highly Adaptive**

- Requirements are specified during initiation and planning
- Risk and cost are controlled by detailed planning based on in-depth analysis of requirements and constraints prior to development
- Key stakeholders are involved at scheduled milestones

- Requirements are elaborated at periodic intervals during software development
- Risk and cost are controlled by progressively detailed planning based on timely specification of requirements and constraints during development
- Key stakeholders are involved at specified intervals

- Requirements are elaborated at frequent intervals during software development
- Risk and cost controlled as requirements and constraints emerge
- Key stakeholders are continuously involved

The Ten Knowledge Areas in PMBOK and SWX

Integration Management
Scope Management
Time Management
Cost Management
Quality Management
Human Resource Management
Communications Management
Risk Management
Procurement Management
Stakeholder Management

The Life Cycle Continuum is not a thin straight line
All SDLCs are multi-dimensional
The processes in each KA must be tailored for the chosen SDLC

Four Orthogonal Concepts: BUP vs BPD vs ID vs IC

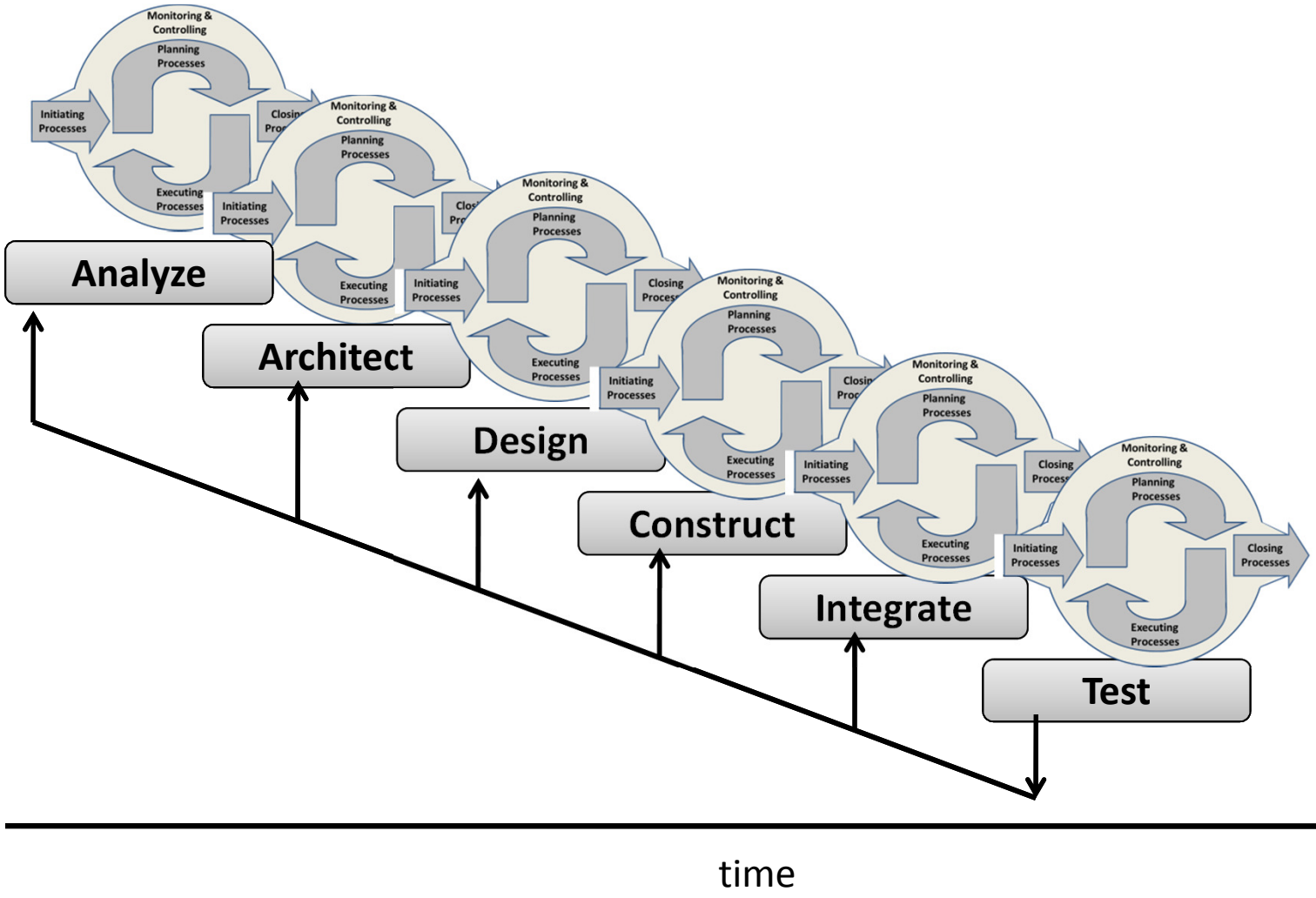
- A predictive life cycle project may involve Big Upfront Planning (BUP)
  - but can avoid Big Product Delivery (BPD)
    - by doing Incremental Development (ID)
  - to produce increments of working software that can be demo'd and delivered into the operational environment
- Iterative Cycles (IC) can be used across the continuum
  - to produce deliverable increments on some iterations
  - to produce internal-demo increments on other iterations
  - to produce no increments on still other iterations
- ID may or may not use IC

A Primary Contribution of SWX

- A primary contribution of SWX is tailoring and adaptation of the 47 processes in the PMBOK Guide for adaptive software project life cycles
  - because many of the processes in the PMBOK Guide are applicable to management of predictive life cycle software projects
    - and are referenced but not repeated in SWX
  - and because the PMBOK Guide includes a discussion of "adaptive" but merges IC with ID
  - and does not discuss tailoring of the 47 processes for different life cycles

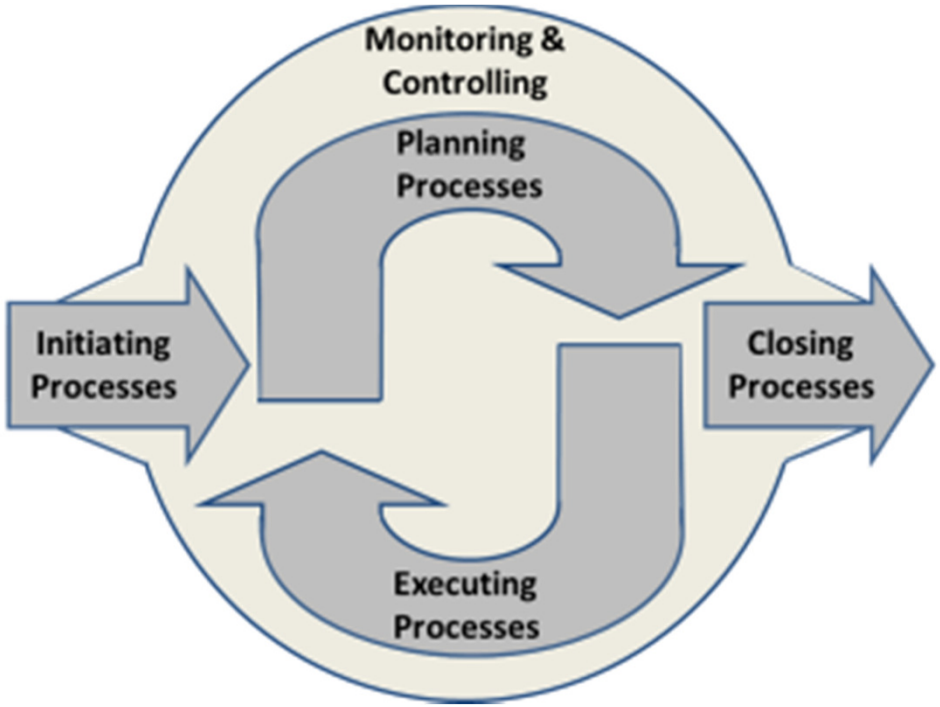
S2EA

A Predictive Software Project Life Cycle

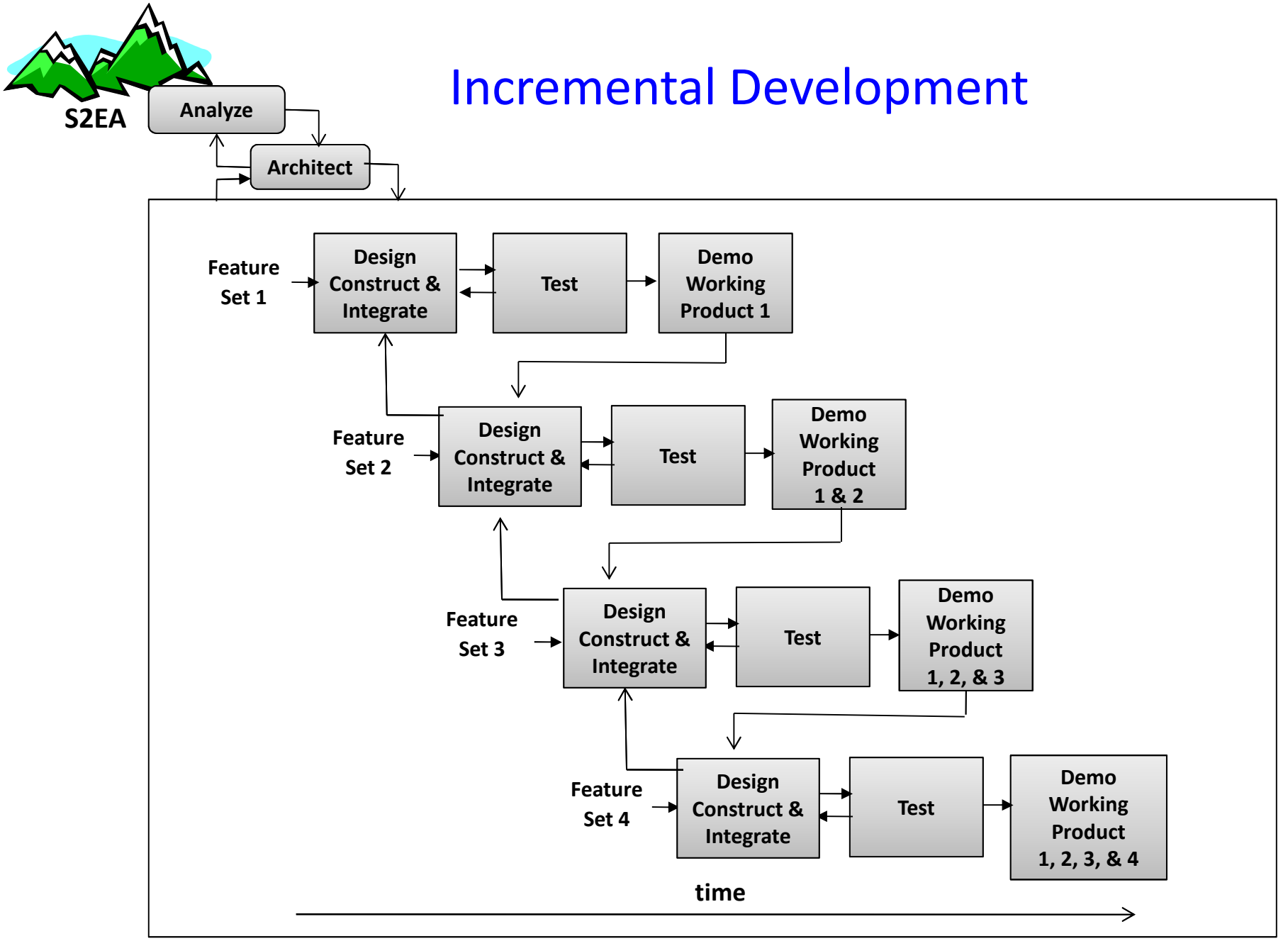


S2EA

The Five Process Groups of the PMBOK® Guide



Incremental Development



Adaptability versus Agility

- “agile” is an overloaded term
  - and is not used in SWX
  - Various adaptive life cycles include various aspects of agility
  - aspects of agility are listed and referred to throughout SWX

Aspects of Agility in Adaptive SDLCs

- Increments of working deliverable software are produced on some iteration cycles
  - increments and iterations are distinct concepts
- Durations of adaptive iteration cycles vary from daily to weekly to monthly, but usually not more than monthly
- Adaptive iteration cycles are often of the same duration (i.e., are “time boxed”) but some cycles may be of longer or shorter duration by exception
- Requirements, design, and the software product emerge as the project evolves

Aspects of Agility in Adaptive SDLCs (2)

- Multi-stage Iterations
  - Iterations can incorporate as many software development stages as desired (from Analyze to Test)
- Vertical Slices
  - increments of delivered functionality include as many architecture components as desired

Aspects of Agility in Adaptive SDLCs (3)

- Adaptive software development teams are small
  - 10 or fewer members
  - large projects include multiple small teams
- All members of each software development team are assigned to one project at a time
- Each software development team includes the generalists and specialists needed to develop the product (cross-functional teams)
  - experts (functional and domain) may be involved periodically or as needed

Adaptive software teams are self-organizing and self-directed

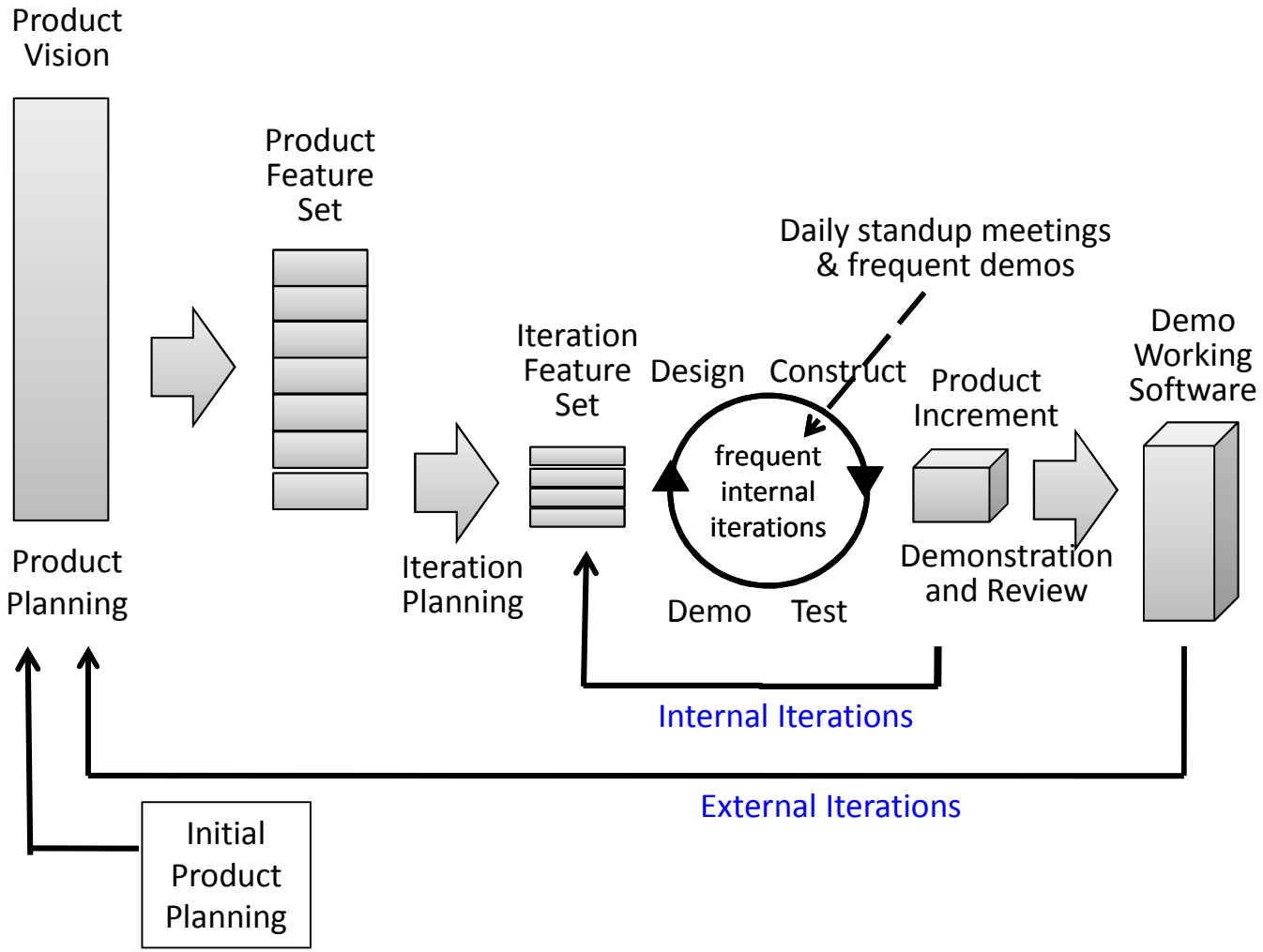
Aspects of Agility in Adaptive SDLCs (4)

- A customer, customer's representative, and/or knowledgeable user is involved on a continuing basis
  - observes periodic demonstrations of working deliverable software (on a daily, weekly, bi-weekly, or monthly basis)
  - provides guidance for further product development

The customer is responsible for project and product scope

S2EA

A Pattern for Adaptive Software Development Life Cycles

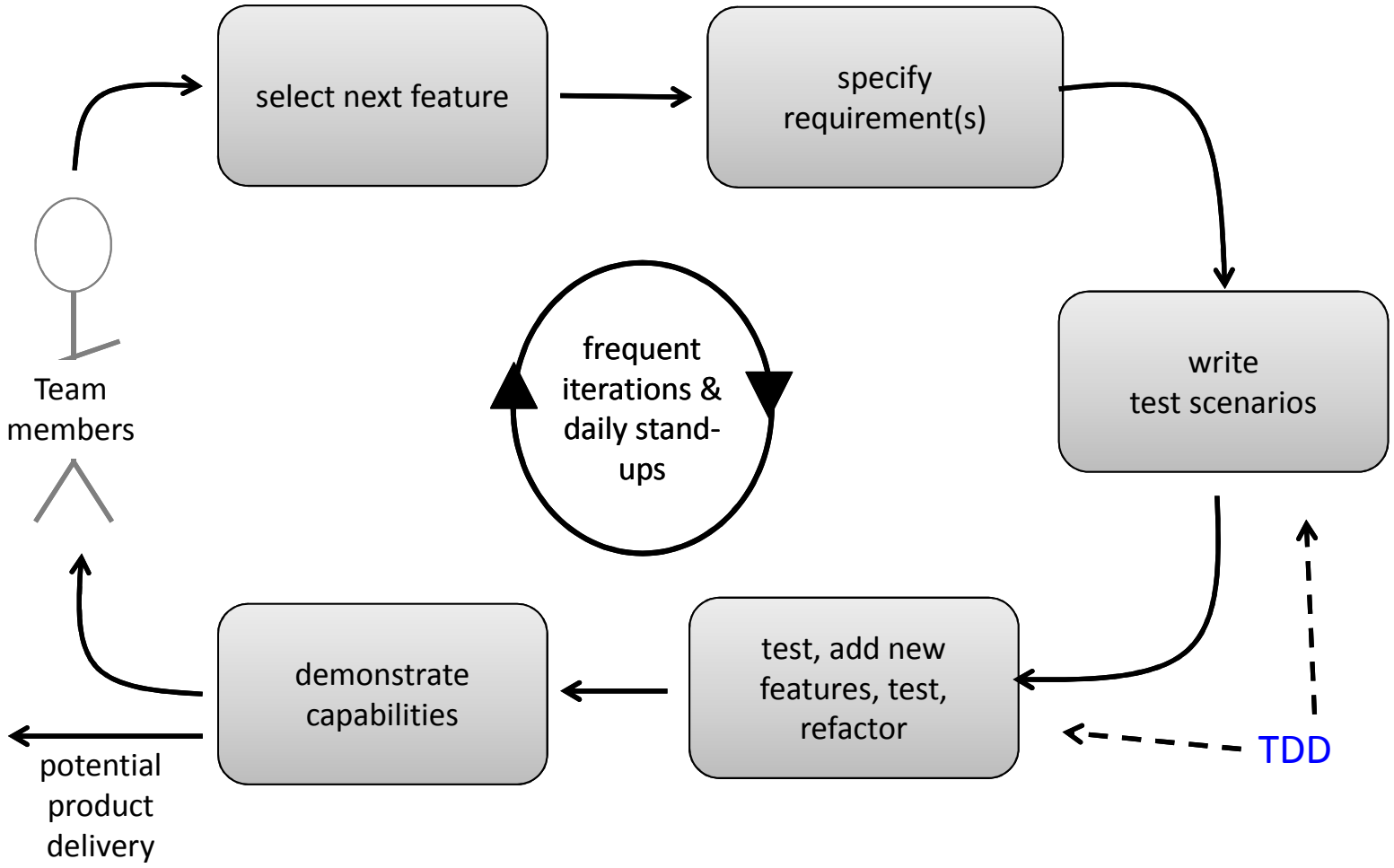


A Quote from SWX (for the previous figure)

"A generic example of a software development method for an adaptive software project life cycle is illustrated in Figure 2-5. This is a common software development pattern, often used as a basis for agile development methods.

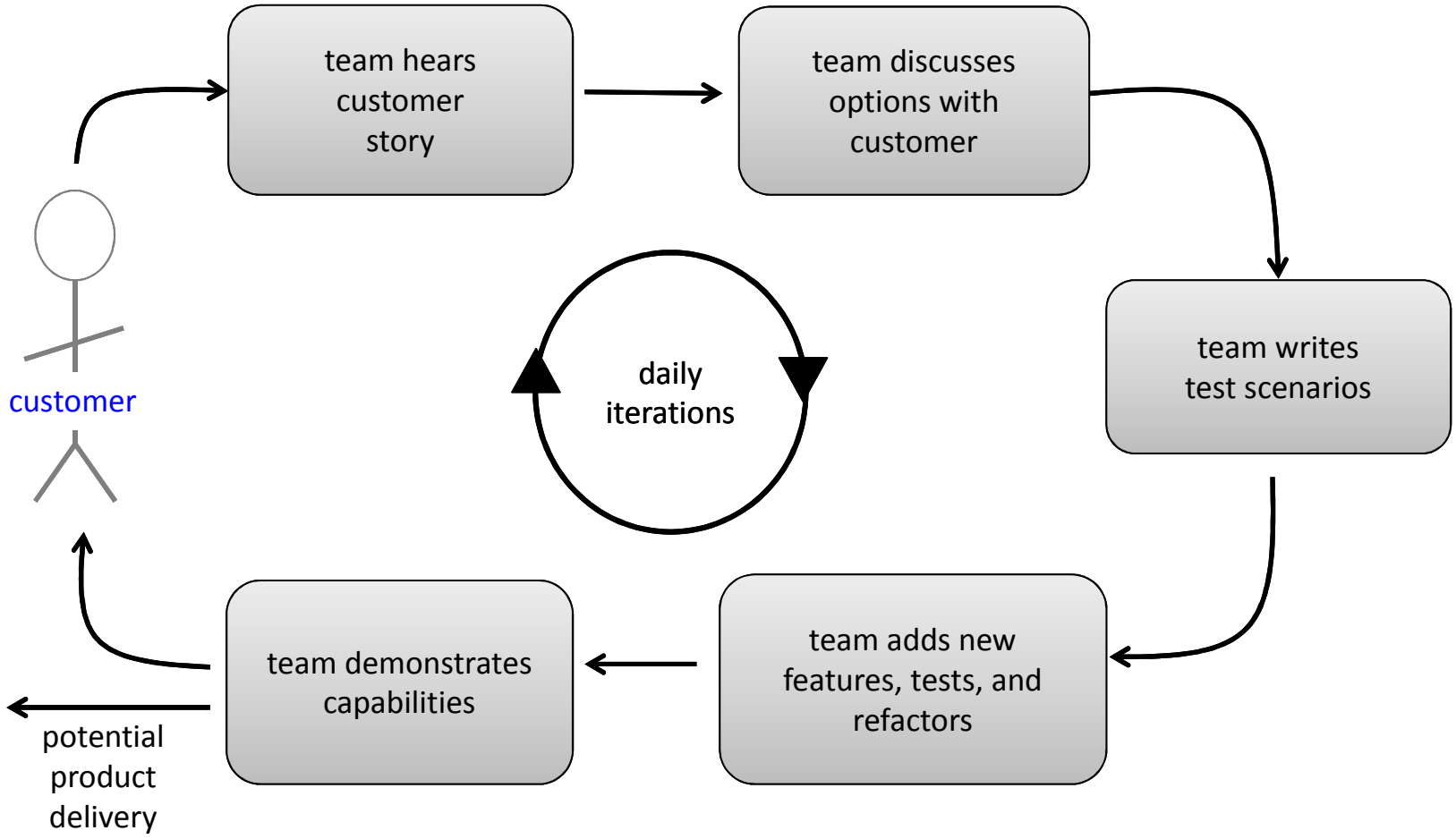
Examples that use variations of this pattern include Scrum, eXtreme Programming, Feature-Driven Development, Test-Driven Development, and the Dynamic System Development Method."

Internals of an Adaptive SDLC with Test-Driven Development (TDD)



S2EA

A Highly Adaptive SDLC



Typical practices of internal adaptive and external highly adaptive software projects

Internal Adaptive Life Cycle	**External Highly Adaptive Life Cycle**
Team members "in the loop"	Customer "in the loop"
Daily team standup meetings and internal demos	Daily demos for the customer
Team selects feature or features for the next internal iteration	Customer supplies story for the next iteration
Team accepts or revises added features	Customer accepts, request revisions, or rejects added features
Software increments available for delivery into the user environment at predetermined intervals, if desired (1, 2, or 4 weeks)	Software increments available for delivery into the user environment at daily intervals, if desired

The SWX Process (1)

- Proposal submitted to the Computer Society and PMI: July 2010
  - lawyers argue
- SWX Committee formed: August 2011
- SME draft prepared: August 2011 – July 2012
- SME draft review: July 2012
  - 27 SME respondents; 730 recommendations
- Public exposure draft prepared: August – November 2012
- Exposure draft review: 10 December 2012 – 10 January 2013
  - 170 respondents; 1973 recommendations
- Individual and team consensus reviews: 11 January – 25 March 2013

The SWX Process (2)

- Reviewers notified: 25 March 2013
- Reviewer's appeals received: 1 April 2013
- Rewrite period: 25 March – 8 April 2013
- Copy editing: 9 April – 7 May 2013
- Committee reviews/ballots: 7 May – 21 May 2013
- PMI & Computer Society review and approval:
  - June – August 2013
- Publication date:
  - electronic: August 2013; print: September 2013

Lessons Learned

- It's always more work than you think it will be
- Avoiding personal ownership of written material is hard
  - egoless writing?
- Good mix of team skills (cross functional) was important
- Real world experience of team members was essential
- Partnership between CS and PMI working well
- Four f2f meetings at six month intervals was good
  - a couple closer in beginning would have helped the team to jell faster
  - f2f meetings contributed to a collaborative learning experience
  - and resulted in a better product

Questions?