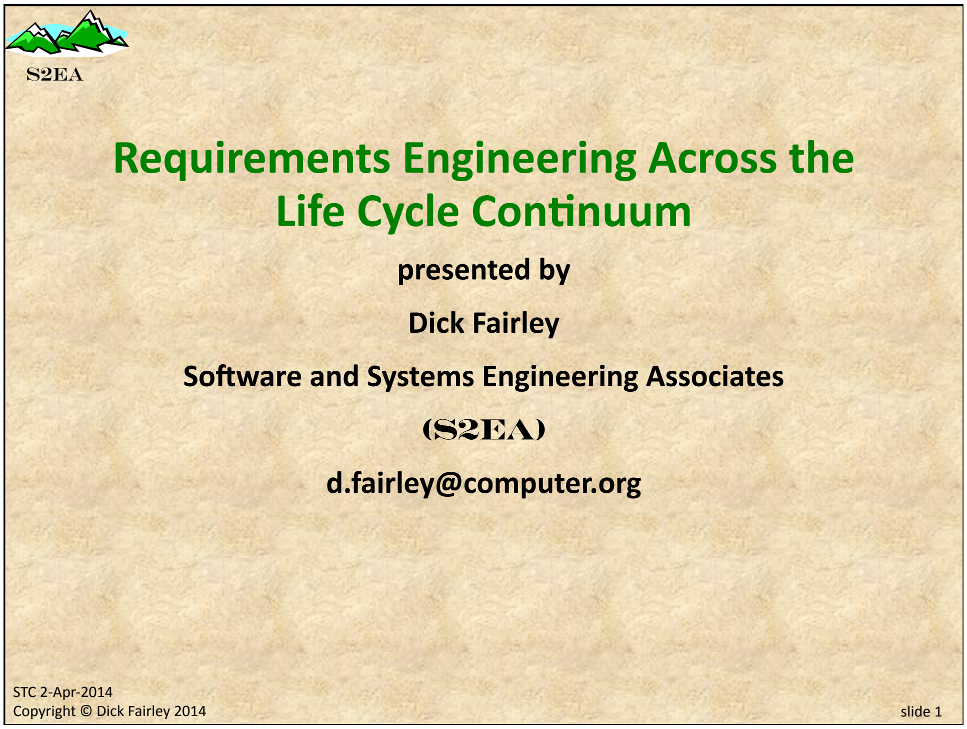


S2EA

Requirements Engineering Across the Life Cycle Continuum

presented by

Dick Fairley

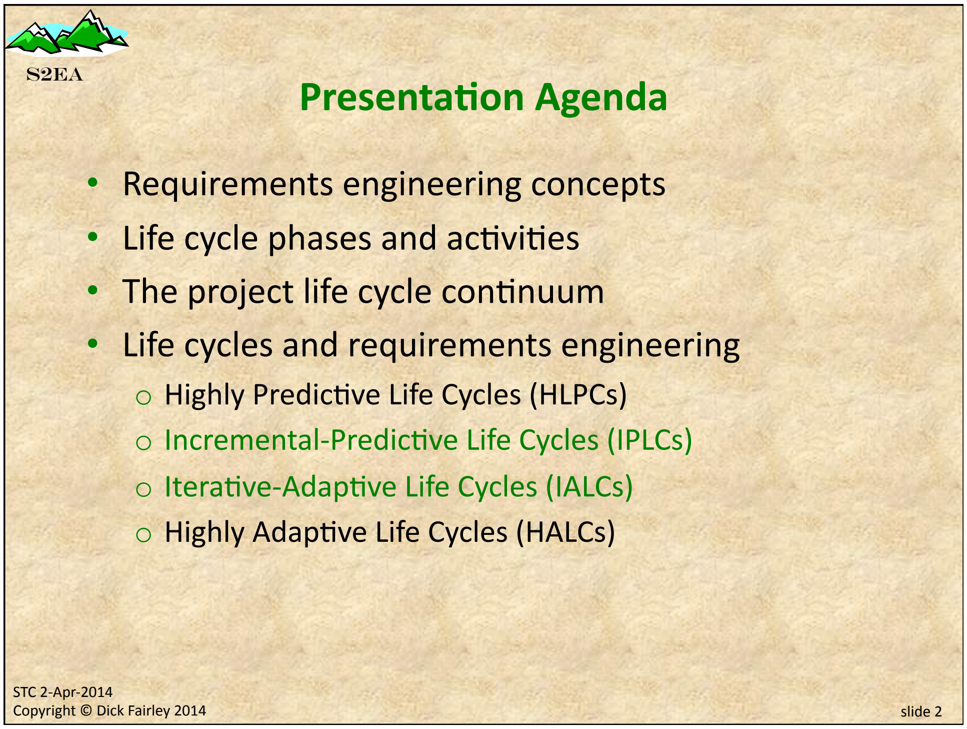
Software and Systems Engineering Associates

(S2EA)

d.fairley@computer.org

STC 2-Apr-2014

slide 1



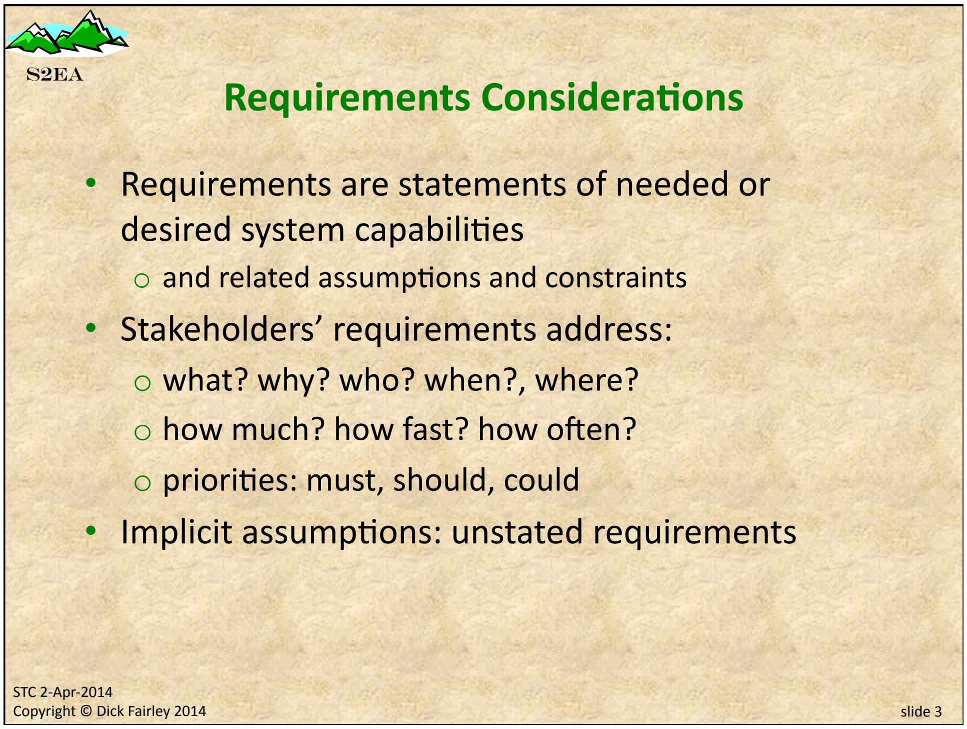
S2EA

Presentation Agenda

- Requirements engineering concepts
- Life cycle phases and activities
- The project life cycle continuum
- Life cycles and requirements engineering
  - Highly Predictive Life Cycles (HLPCs)
  - Incremental-Predictive Life Cycles (IPLCs)
  - Iterative-Adaptive Life Cycles (IALCs)
  - Highly Adaptive Life Cycles (HALCs)

STC 2-Apr-2014

slide 2



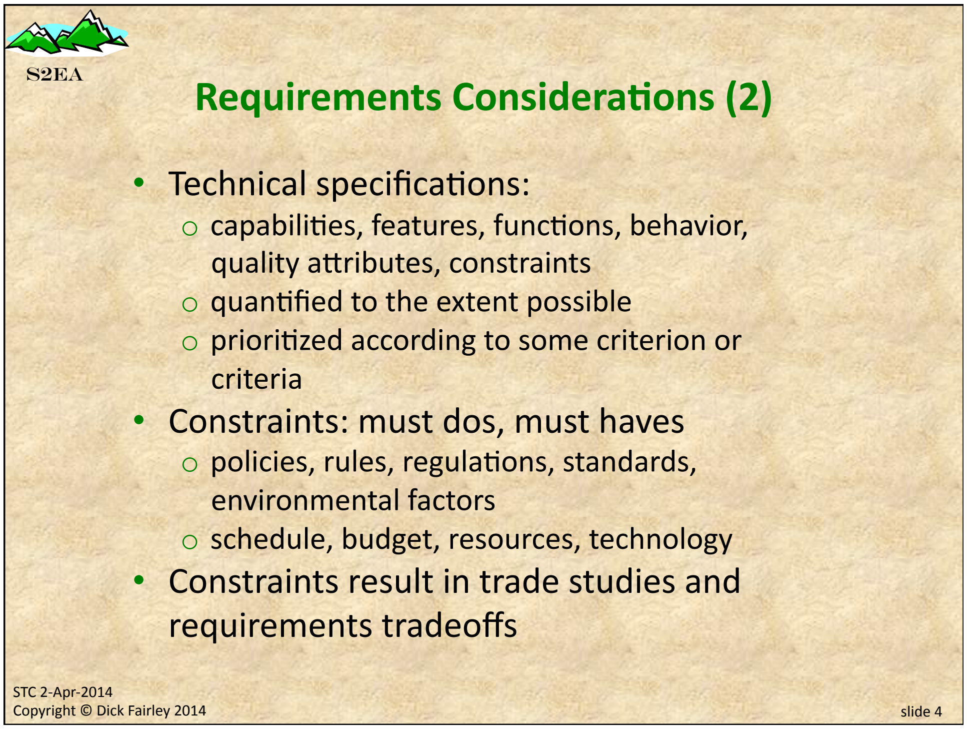
S2EA

Requirements Considerations

- Requirements are statements of needed or desired system capabilities
  - ○ and related assumptions and constraints
- Stakeholders' requirements address:
  - ○ what? why? who? when?, where?
  - ○ how much? how fast? how often?
  - ○ priorities: must, should, could
- Implicit assumptions: unstated requirements

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 3



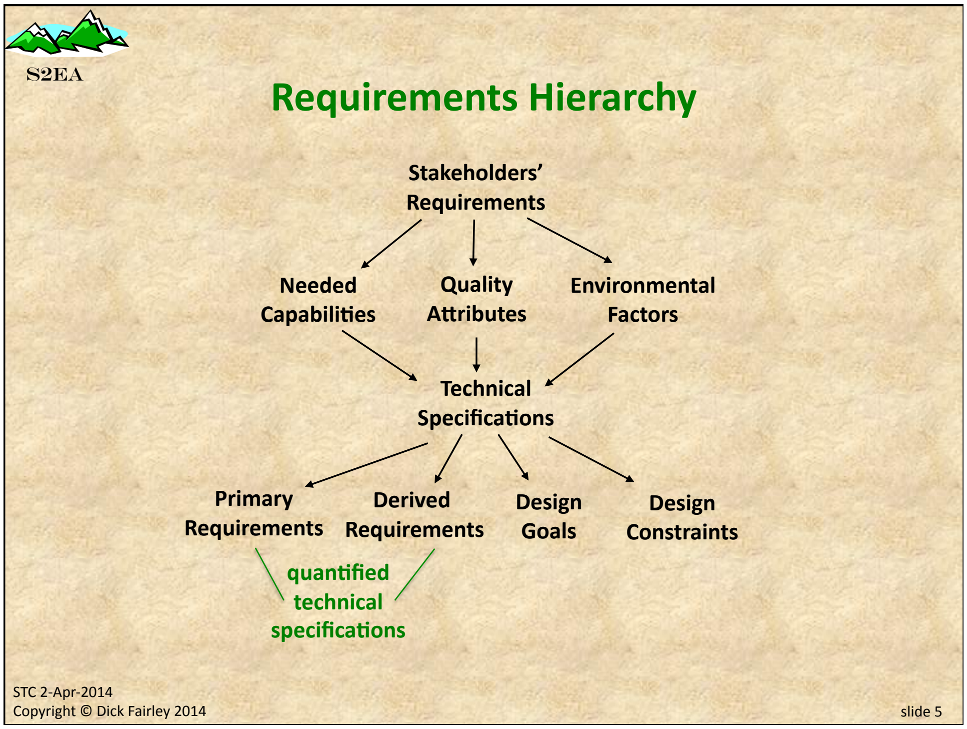
S2EA

Requirements Considerations (2)

- Technical specifications:
  - ○ capabilities, features, functions, behavior, quality attributes, constraints
  - ○ quantified to the extent possible
  - ○ prioritized according to some criterion or criteria
- Constraints: must dos, must haves
  - ○ policies, rules, regulations, standards, environmental factors
  - ○ schedule, budget, resources, technology
- Constraints result in trade studies and requirements tradeoffs

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 4



S2EA

Requirements Hierarchy

Stakeholders' Requirements

Needed Capabilities

Quality Attributes

Environmental Factors

Technical Specifications

Primary Requirements

Derived Requirements

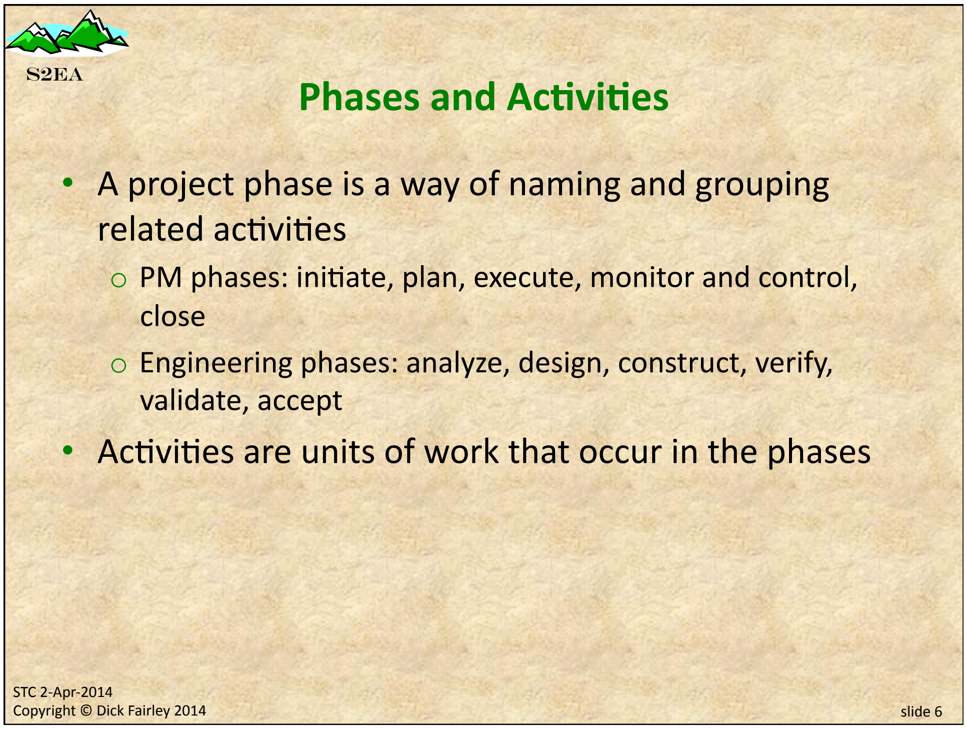
Design Goals

Design Constraints

quantified technical specifications

STC 2-Apr-2014

slide 5



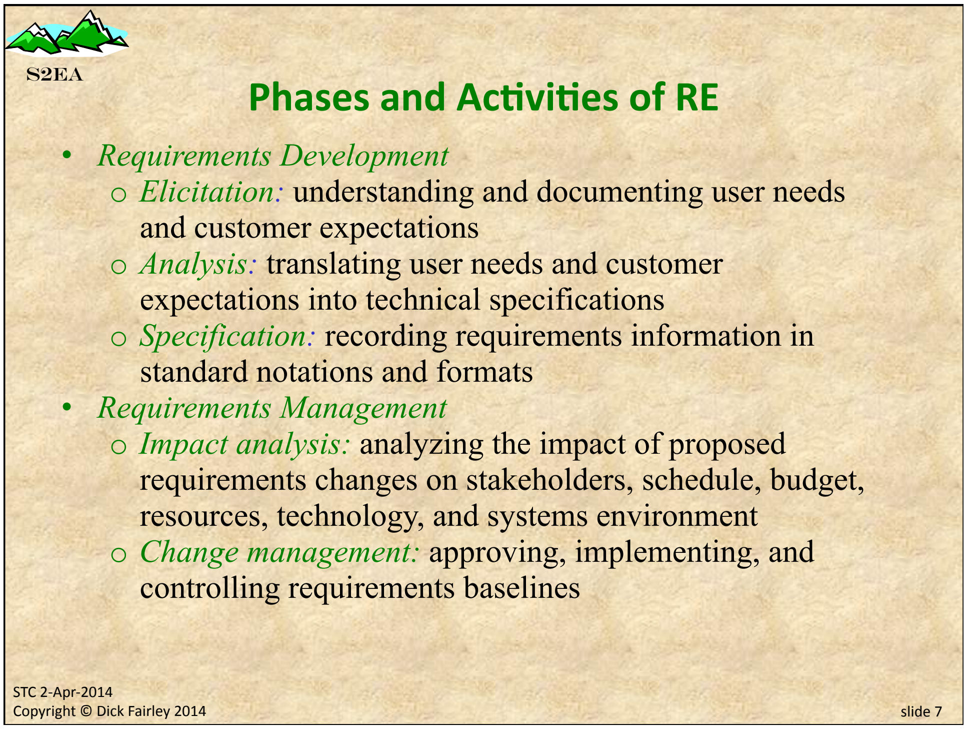
S2EA

Phases and Activities

- A project phase is a way of naming and grouping related activities
  - PM phases: initiate, plan, execute, monitor and control, close
  - Engineering phases: analyze, design, construct, verify, validate, accept
- Activities are units of work that occur in the phases

STC 2-Apr-2014

slide 6



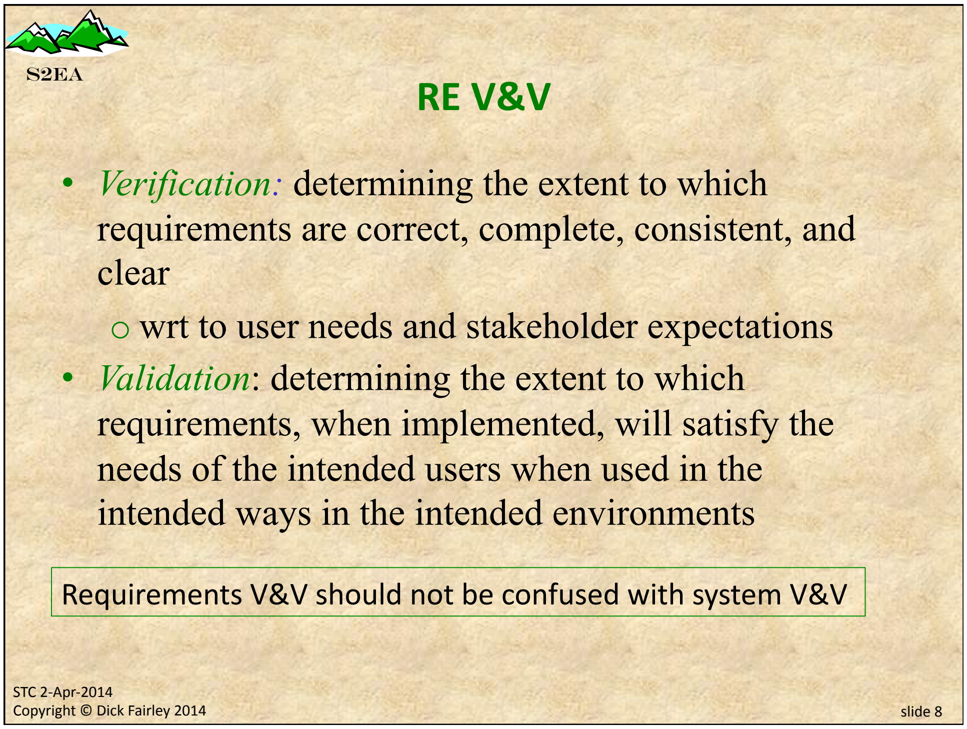
S2EA

Phases and Activities of RE

- *Requirements Development*
  - *Elicitation:* understanding and documenting user needs and customer expectations
  - *Analysis:* translating user needs and customer expectations into technical specifications
  - *Specification:* recording requirements information in standard notations and formats
- *Requirements Management*
  - *Impact analysis:* analyzing the impact of proposed requirements changes on stakeholders, schedule, budget, resources, technology, and systems environment
  - *Change management:* approving, implementing, and controlling requirements baselines

STC 2-Apr-2014

slide 7



S2EA

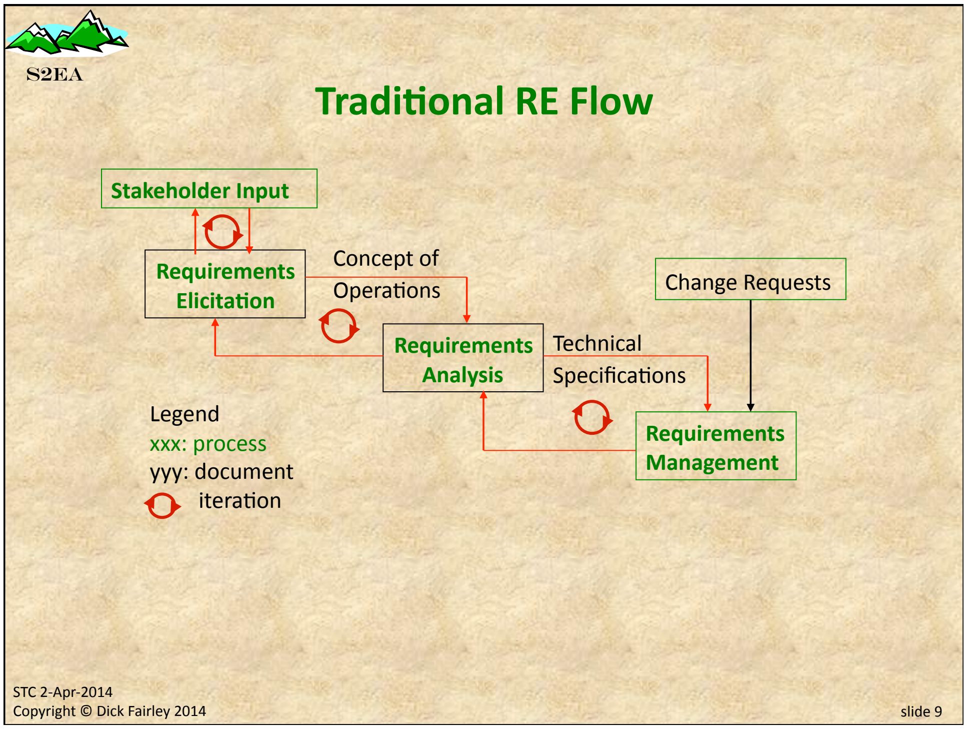
RE V&V

- *Verification:* determining the extent to which requirements are correct, complete, consistent, and clear
  - wrt to user needs and stakeholder expectations
- *Validation*: determining the extent to which requirements, when implemented, will satisfy the needs of the intended users when used in the intended ways in the intended environments

Requirements V&V should not be confused with system V&V

STC 2-Apr-2014

slide 8



S2EA

Traditional RE Flow

Stakeholder Input

Requirements Elicitation

Concept of Operations

Change Requests

Requirements Analysis

Technical Specifications

Requirements Management

Legend

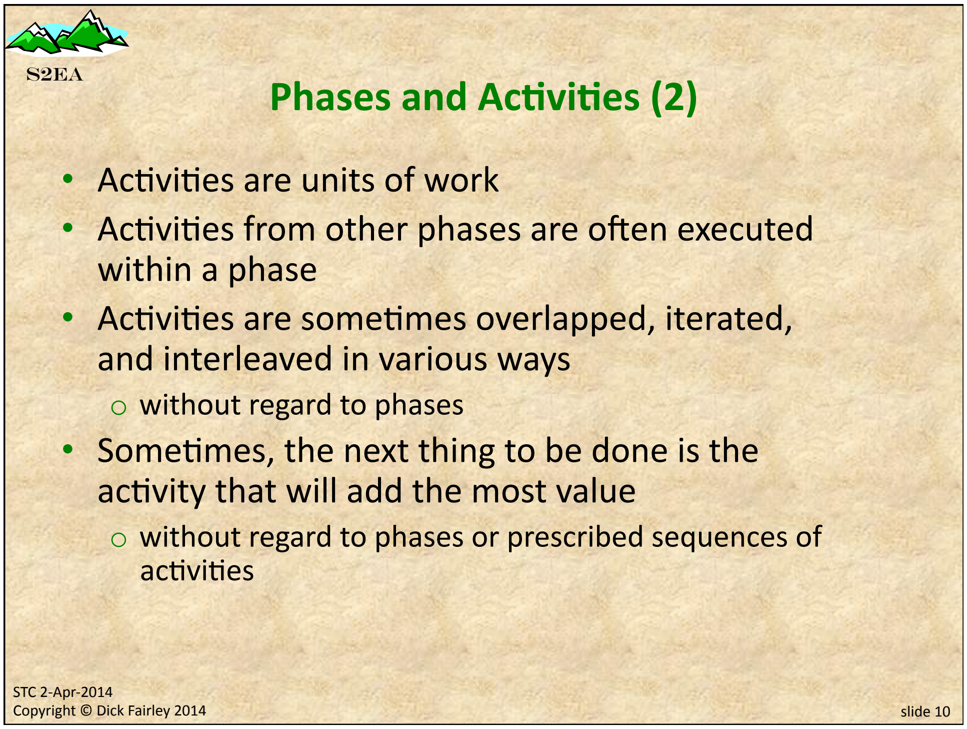
xxx: process

yyy: document

⟳ iteration

STC 2-Apr-2014

slide 9



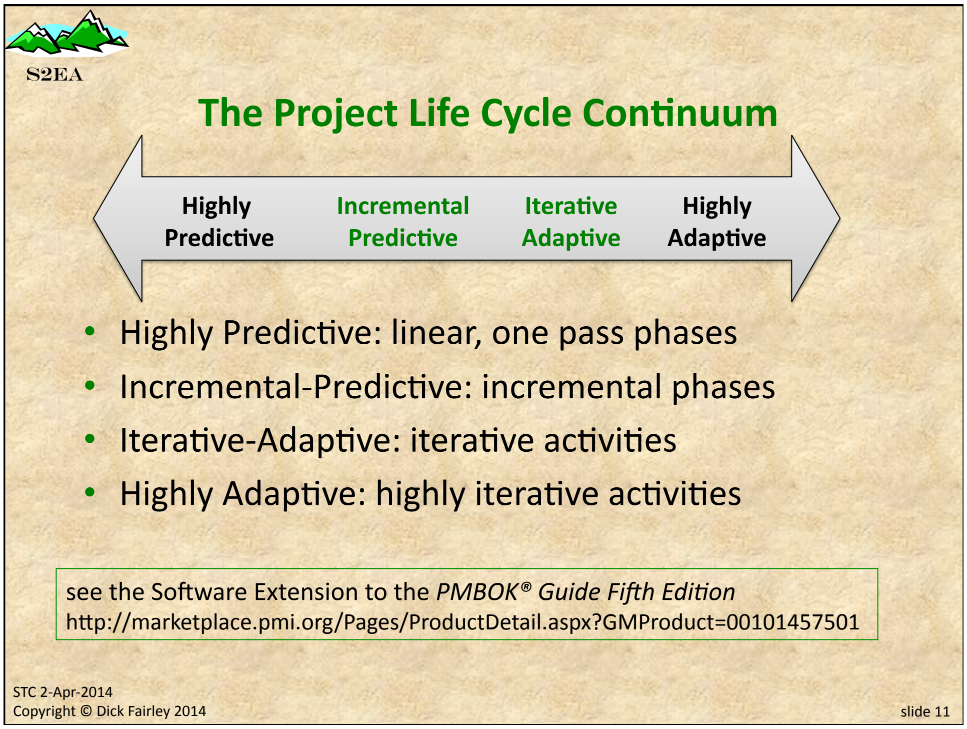
S2EA

Phases and Activities (2)

- Activities are units of work
- Activities from other phases are often executed within a phase
- Activities are sometimes overlapped, iterated, and interleaved in various ways
  - without regard to phases
- Sometimes, the next thing to be done is the activity that will add the most value
  - without regard to phases or prescribed sequences of activities

STC 2-Apr-2014

slide 10



S2EA

The Project Life Cycle Continuum

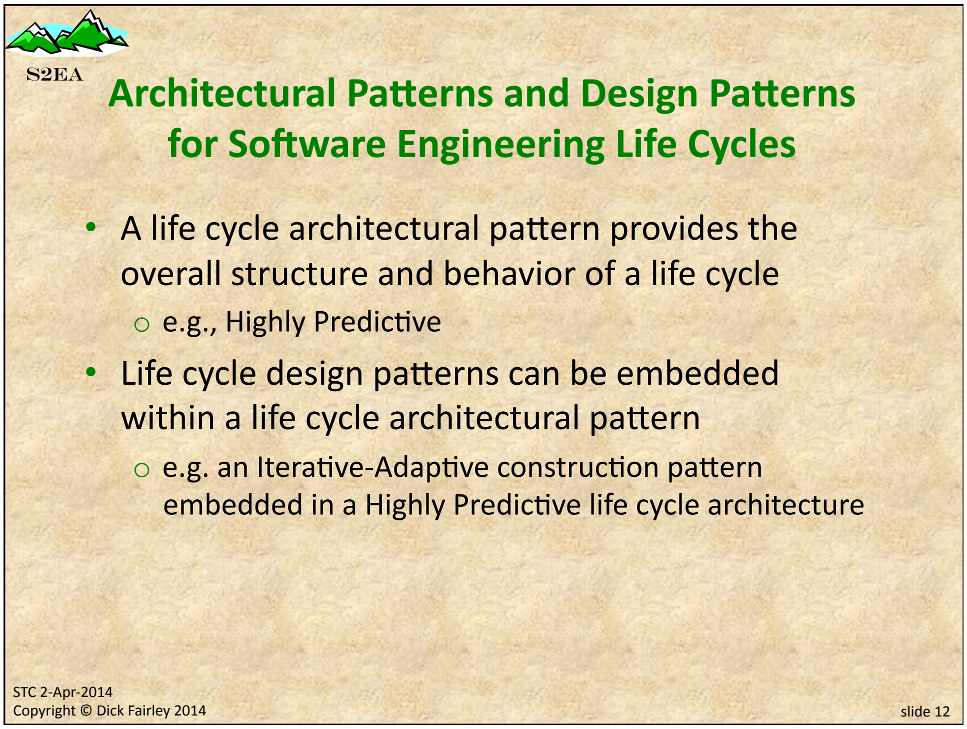
Highly Predictive | Incremental Predictive | Iterative Adaptive | Highly Adaptive

- Highly Predictive: linear, one pass phases
- Incremental-Predictive: incremental phases
- Iterative-Adaptive: iterative activities
- Highly Adaptive: highly iterative activities

see the Software Extension to the *PMBOK® Guide Fifth Edition*
http://marketplace.pmi.org/Pages/ProductDetail.aspx?GMProduct=00101457501

STC 2-Apr-2014

slide 11



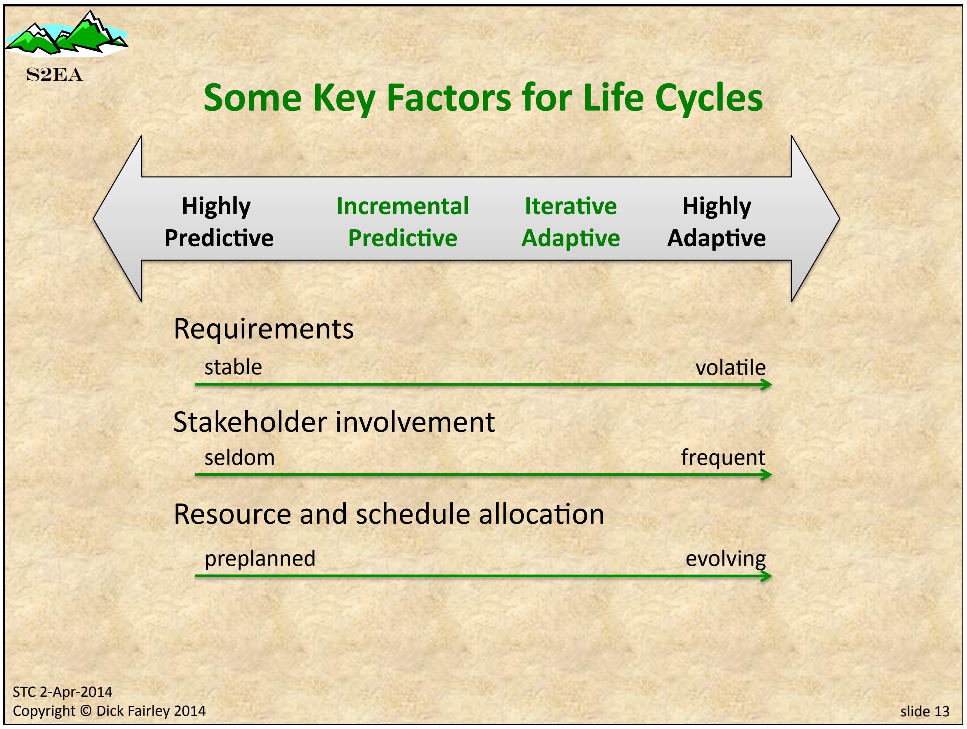
S2EA

Architectural Patterns and Design Patterns for Software Engineering Life Cycles

- A life cycle architectural pattern provides the overall structure and behavior of a life cycle
  - e.g., Highly Predictive
- Life cycle design patterns can be embedded within a life cycle architectural pattern
  - e.g. an Iterative-Adaptive construction pattern embedded in a Highly Predictive life cycle architecture

STC 2-Apr-2014

slide 12



S2EA

Some Key Factors for Life Cycles

Highly Predictive | Incremental Predictive | Iterative Adaptive | Highly Adaptive

Requirements

stable → volatile

Stakeholder involvement

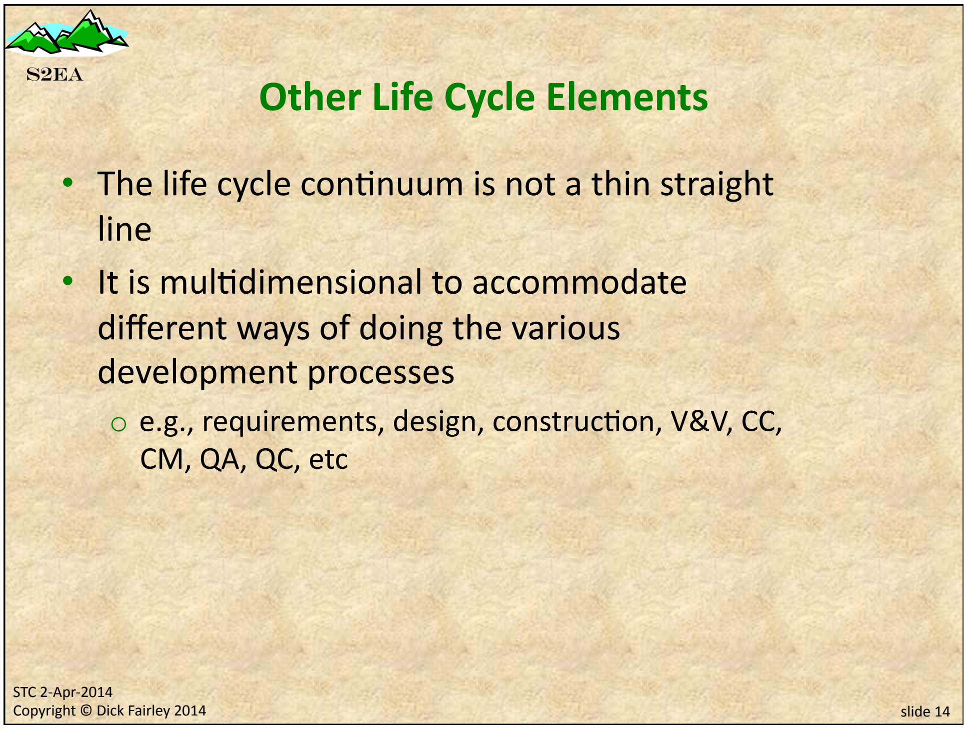
seldom → frequent

Resource and schedule allocation

preplanned → evolving

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 13



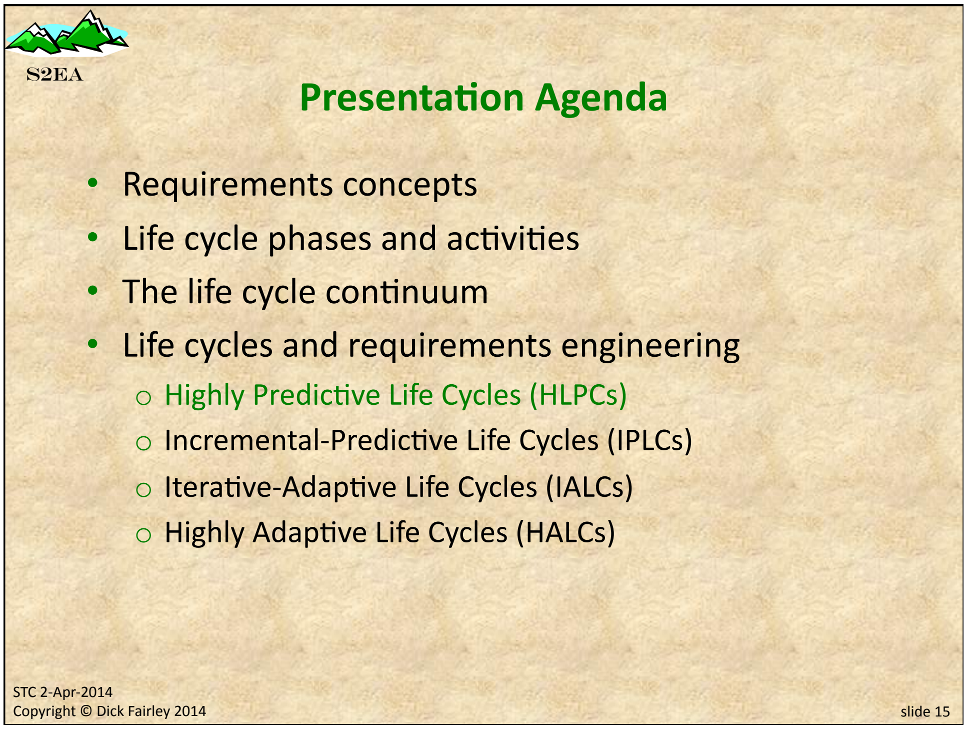
S2EA

Other Life Cycle Elements

- The life cycle continuum is not a thin straight line
- It is multidimensional to accommodate different ways of doing the various development processes
  - e.g., requirements, design, construction, V&V, CC, CM, QA, QC, etc

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 14



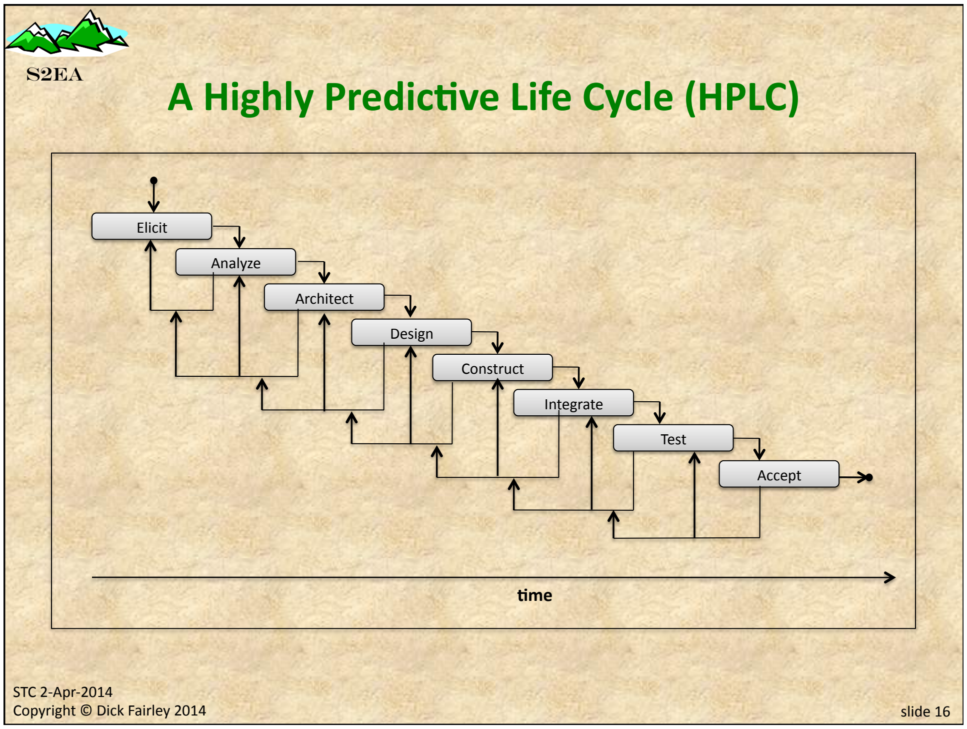
S2EA

Presentation Agenda

- Requirements concepts
- Life cycle phases and activities
- The life cycle continuum
- Life cycles and requirements engineering
  - ○ Highly Predictive Life Cycles (HLPCs)
  - ○ Incremental-Predictive Life Cycles (IPLCs)
  - ○ Iterative-Adaptive Life Cycles (IALCs)
  - ○ Highly Adaptive Life Cycles (HALCs)

STC 2-Apr-2014

slide 15

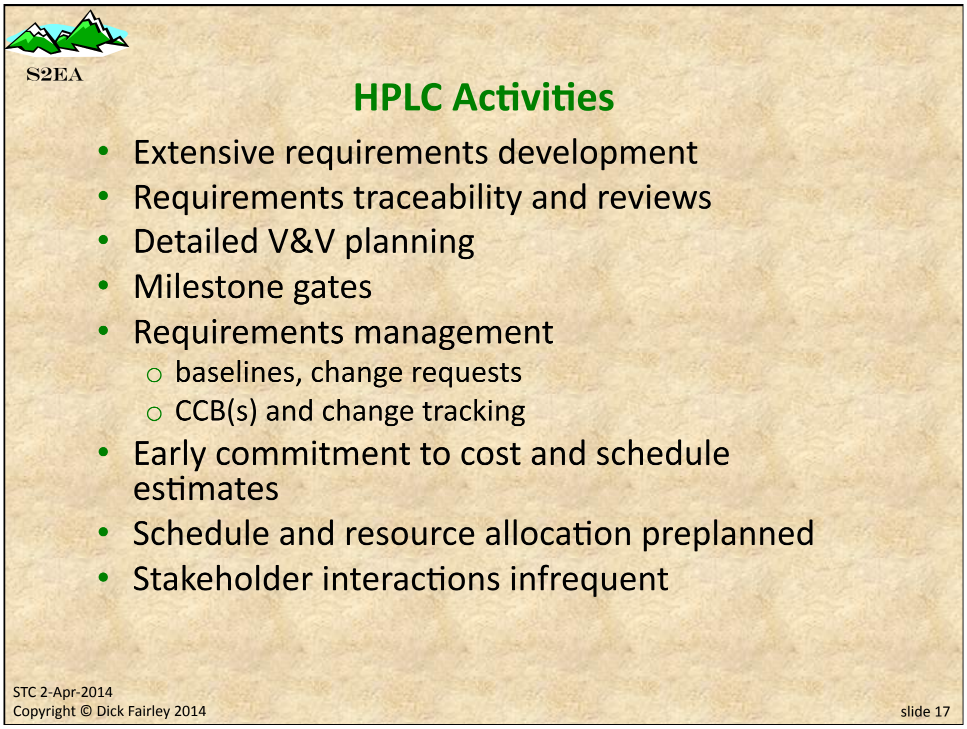


S2EA

A Highly Predictive Life Cycle (HPLC)

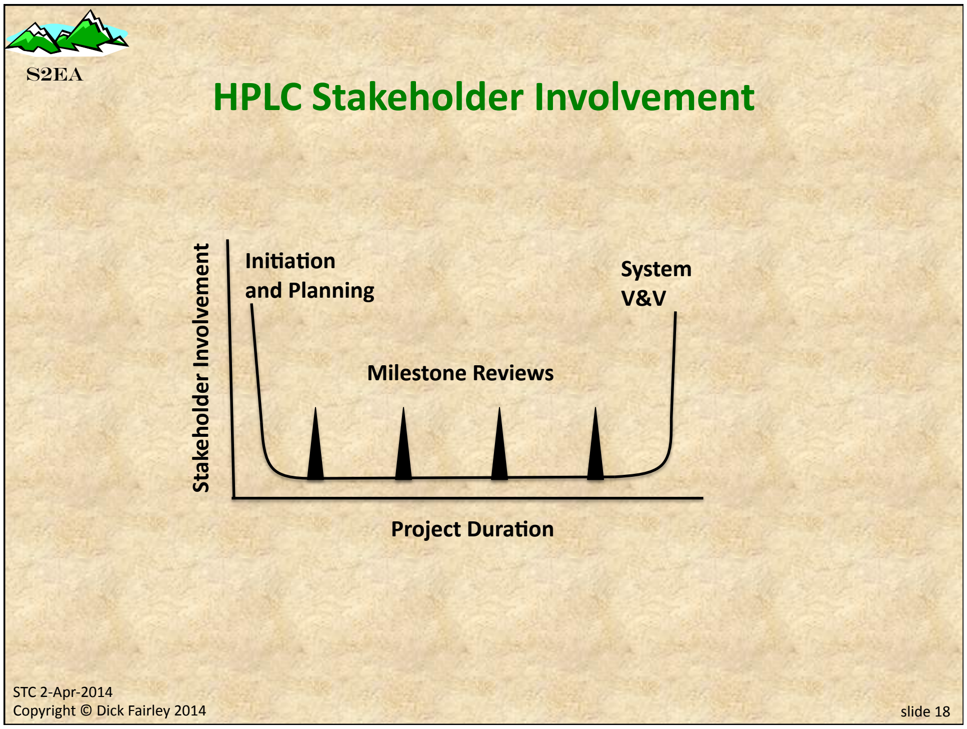
STC 2-Apr-2014

slide 16

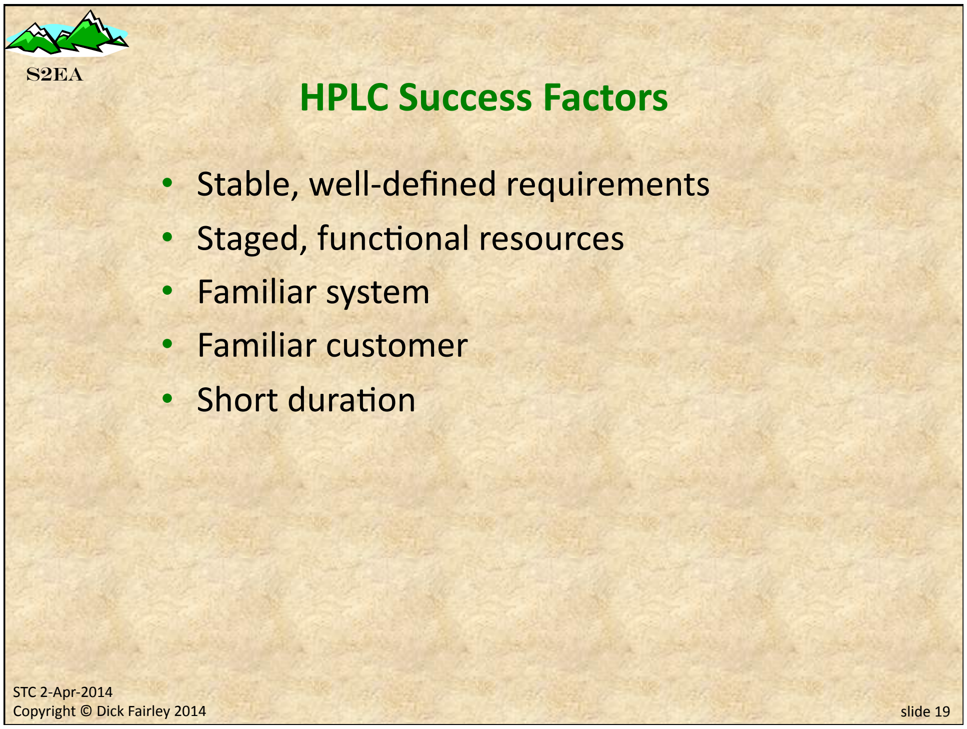


HPLC Activities

- Extensive requirements development
- Requirements traceability and reviews
- Detailed V&V planning
- Milestone gates
- Requirements management
  - baselines, change requests
  - CCB(s) and change tracking
- Early commitment to cost and schedule estimates
- Schedule and resource allocation preplanned
- Stakeholder interactions infrequent



HPLC Stakeholder Involvement



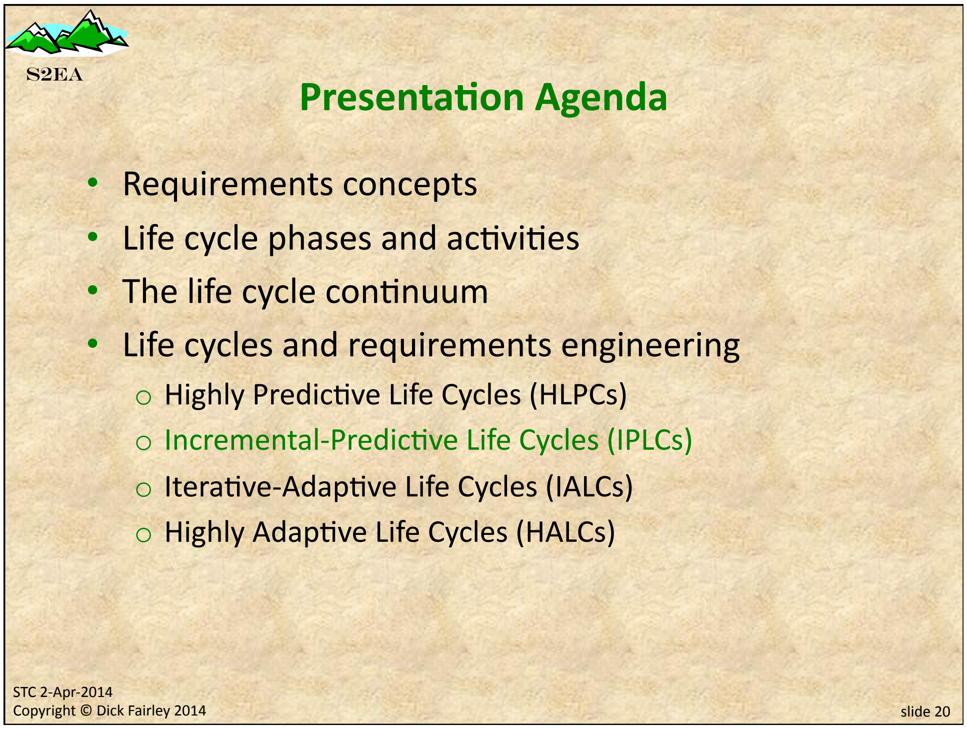
S2EA

HPLC Success Factors

- Stable, well-defined requirements
- Staged, functional resources
- Familiar system
- Familiar customer
- Short duration

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 19



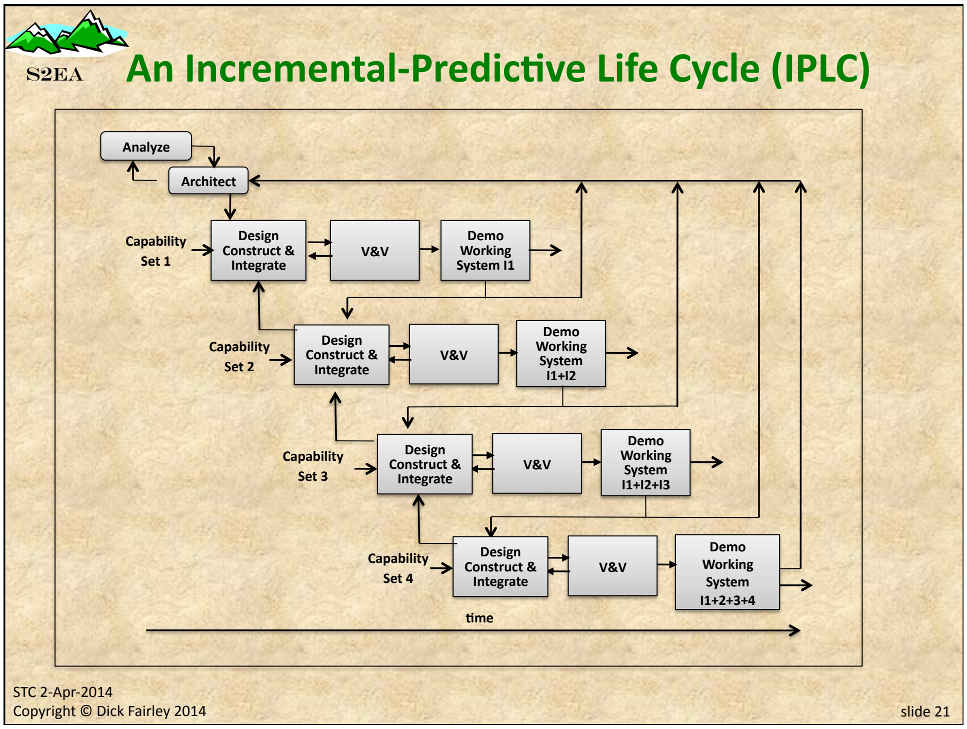
S2EA

Presentation Agenda

- Requirements concepts
- Life cycle phases and activities
- The life cycle continuum
- Life cycles and requirements engineering
  - ○ Highly Predictive Life Cycles (HLPCs)
  - ○ Incremental-Predictive Life Cycles (IPLCs)
  - ○ Iterative-Adaptive Life Cycles (IALCs)
  - ○ Highly Adaptive Life Cycles (HALCs)

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 20



S2EA

An Incremental-Predictive Life Cycle (IPLC)

Analyze

Architect

Capability Set 1 → Design Construct & Integrate → V&V → Demo Working System I1

Capability Set 2 → Design Construct & Integrate → V&V → Demo Working System I1+I2

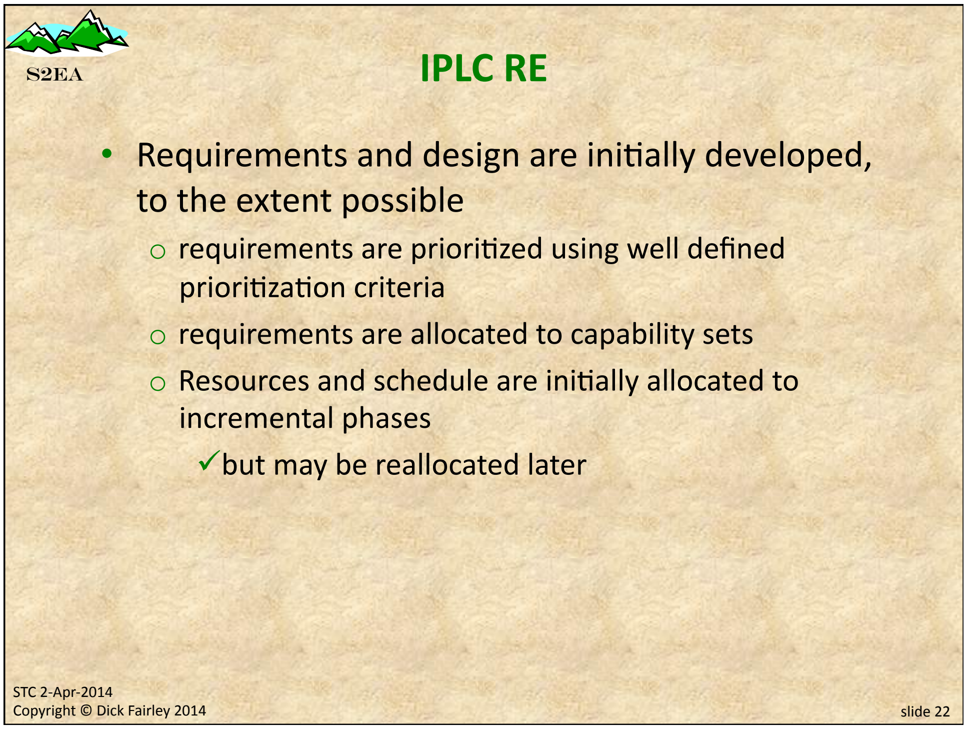
Capability Set 3 → Design Construct & Integrate → V&V → Demo Working System I1+I2+I3

Capability Set 4 → Design Construct & Integrate → V&V → Demo Working System I1+2+3+4

time

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 21



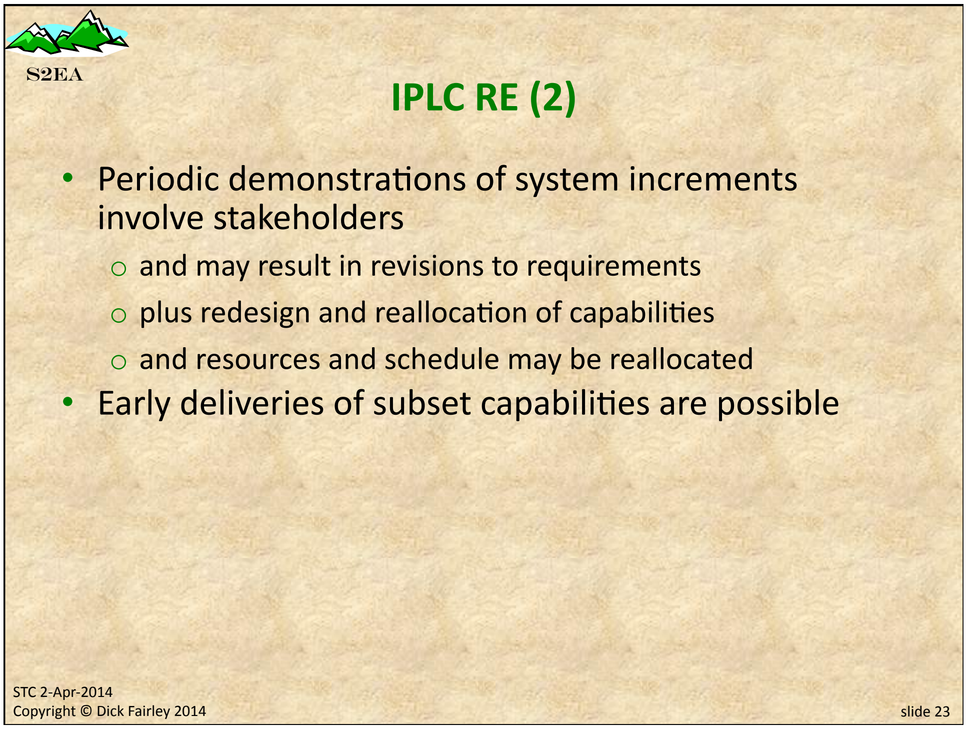
S2EA

IPLC RE

- Requirements and design are initially developed, to the extent possible
  - ○ requirements are prioritized using well defined prioritization criteria
  - ○ requirements are allocated to capability sets
  - ○ Resources and schedule are initially allocated to incremental phases
    - ✓ but may be reallocated later

STC 2-Apr-2014
Copyright © Dick Fairley 2014

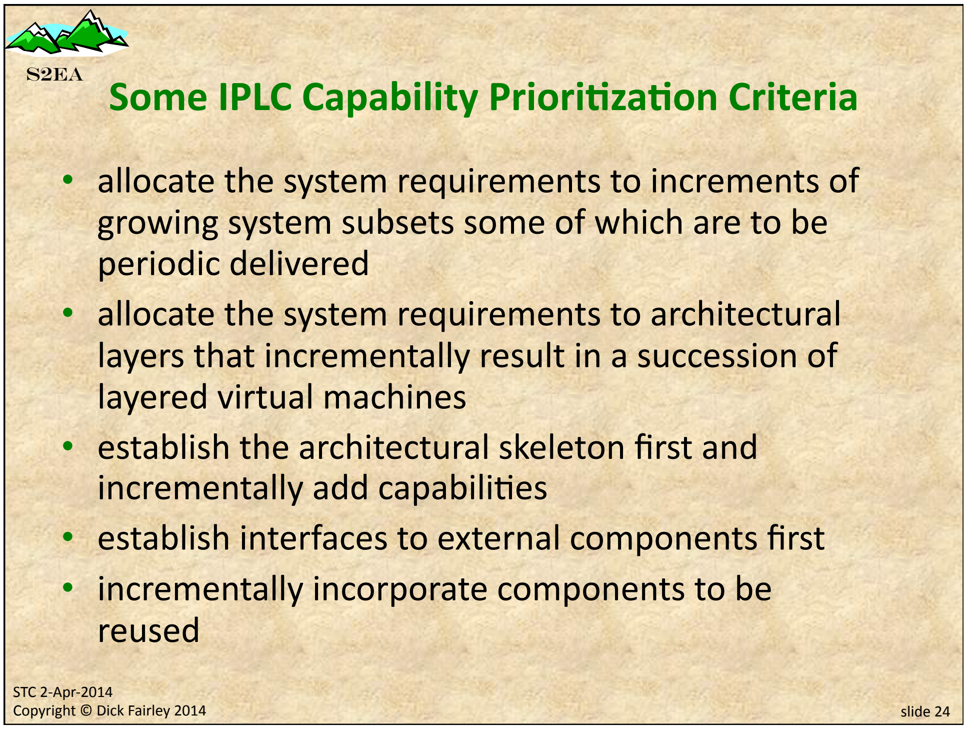
slide 22



S2EA

IPLC RE (2)

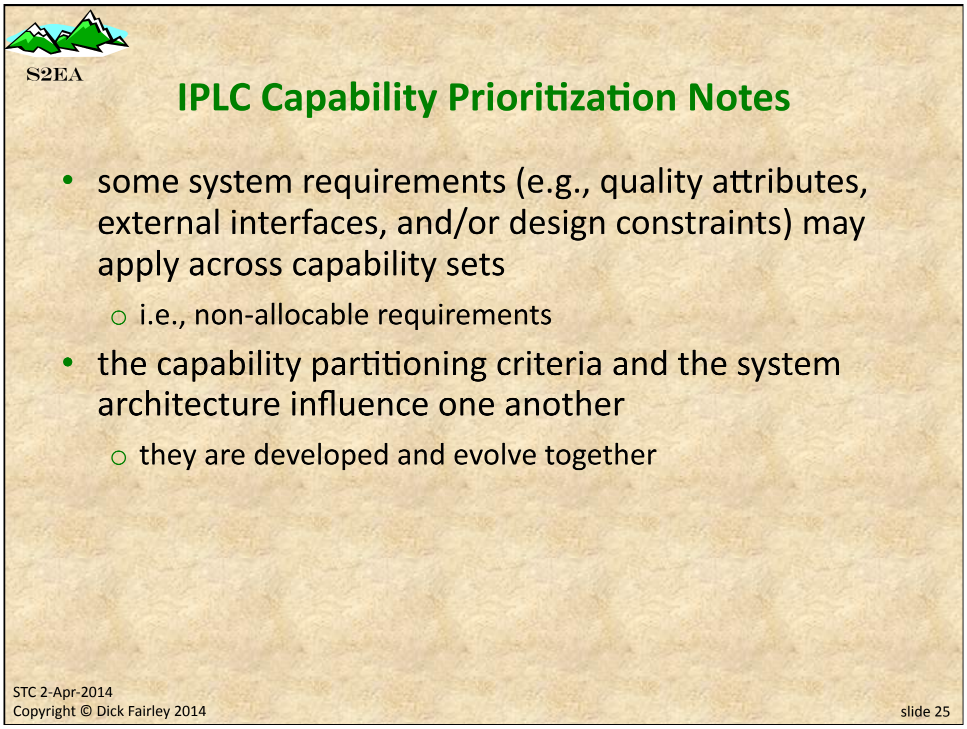
- Periodic demonstrations of system increments involve stakeholders
  - ○ and may result in revisions to requirements
  - ○ plus redesign and reallocation of capabilities
  - ○ and resources and schedule may be reallocated
- Early deliveries of subset capabilities are possible



S2EA

Some IPLC Capability Prioritization Criteria

- allocate the system requirements to increments of growing system subsets some of which are to be periodic delivered
- allocate the system requirements to architectural layers that incrementally result in a succession of layered virtual machines
- establish the architectural skeleton first and incrementally add capabilities
- establish interfaces to external components first
- incrementally incorporate components to be reused

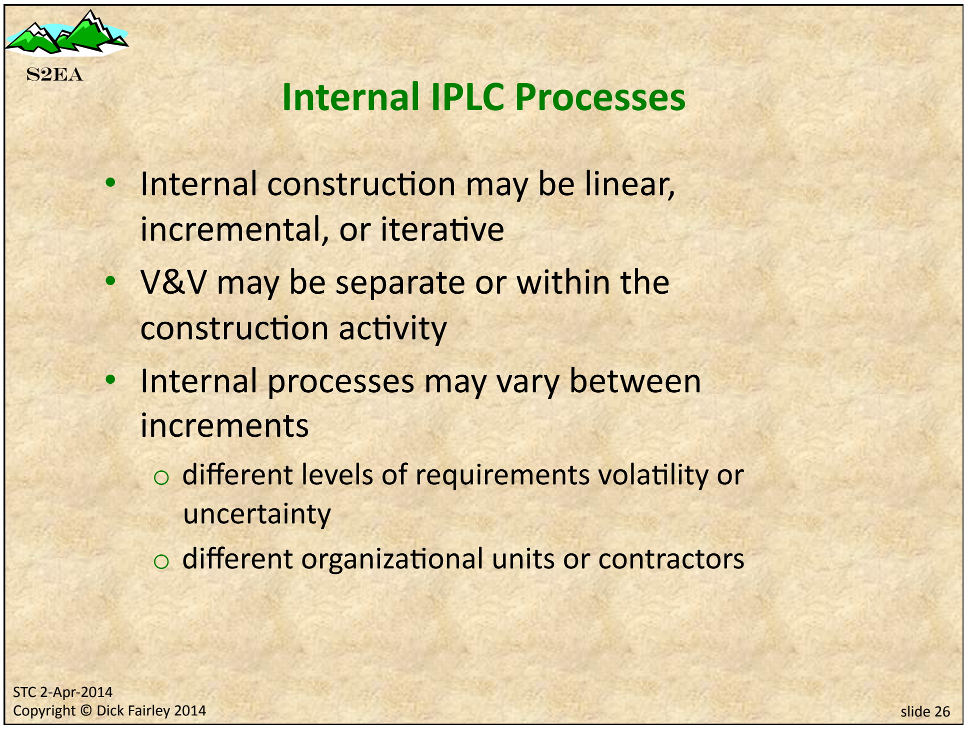


S2EA

IPLC Capability Prioritization Notes

- some system requirements (e.g., quality attributes, external interfaces, and/or design constraints) may apply across capability sets
  - i.e., non-allocable requirements
- the capability partitioning criteria and the system architecture influence one another
  - they are developed and evolve together

STC 2-Apr-2014
Copyright © Dick Fairley 2014
slide 25

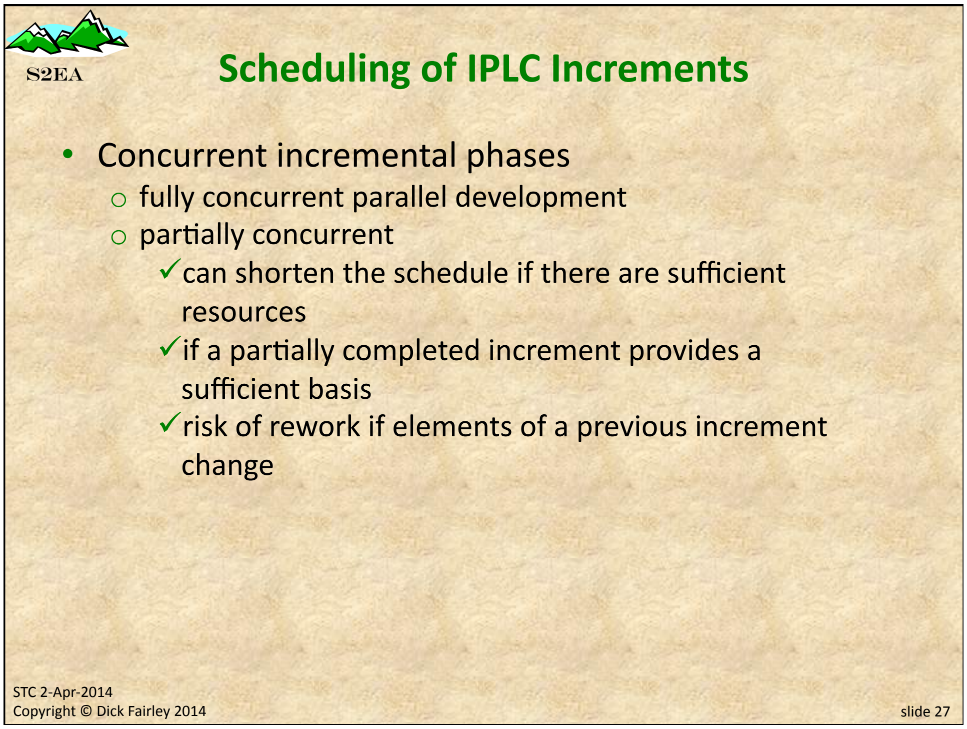


S2EA

Internal IPLC Processes

- Internal construction may be linear, incremental, or iterative
- V&V may be separate or within the construction activity
- Internal processes may vary between increments
  - different levels of requirements volatility or uncertainty
  - different organizational units or contractors

STC 2-Apr-2014
Copyright © Dick Fairley 2014
slide 26



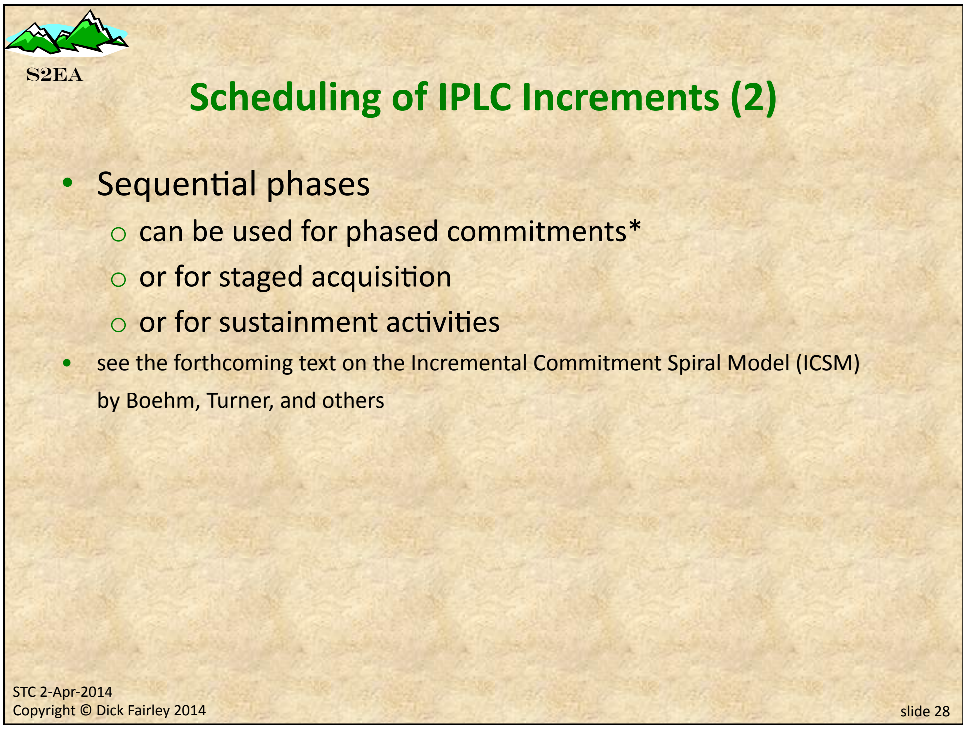
S2EA

Scheduling of IPLC Increments

- Concurrent incremental phases
  - ○ fully concurrent parallel development
  - ○ partially concurrent
    - ✓can shorten the schedule if there are sufficient resources
    - ✓if a partially completed increment provides a sufficient basis
    - ✓risk of rework if elements of a previous increment change

STC 2-Apr-2014

slide 27



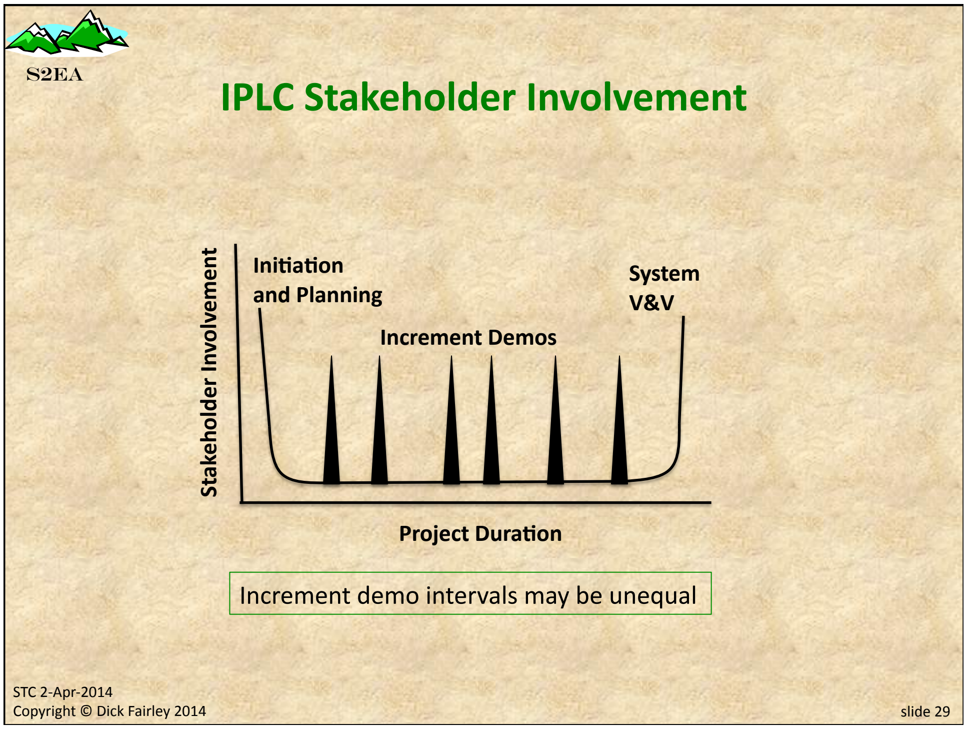
S2EA

Scheduling of IPLC Increments (2)

- Sequential phases
  - ○ can be used for phased commitments*
  - ○ or for staged acquisition
  - ○ or for sustainment activities
- see the forthcoming text on the Incremental Commitment Spiral Model (ICSM) by Boehm, Turner, and others

STC 2-Apr-2014

slide 28



S2EA

IPLC Stakeholder Involvement

Stakeholder Involvement

Initiation and Planning

Increment Demos

System V&V

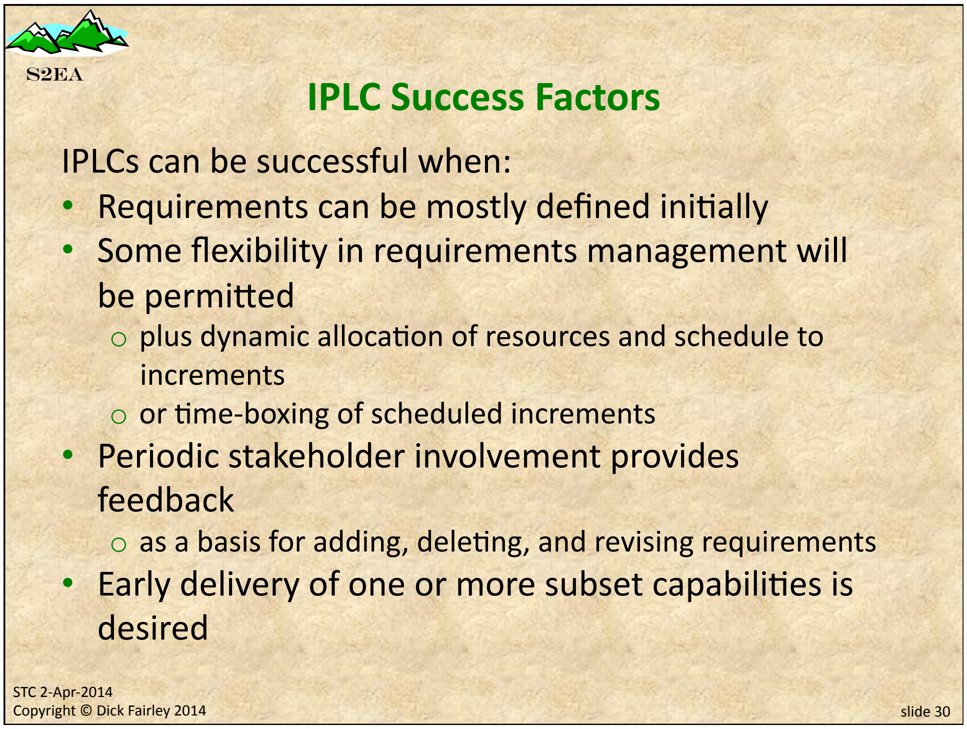
Project Duration

Increment demo intervals may be unequal

STC 2-Apr-2014

Copyright © Dick Fairley 2014

slide 29



S2EA

IPLC Success Factors

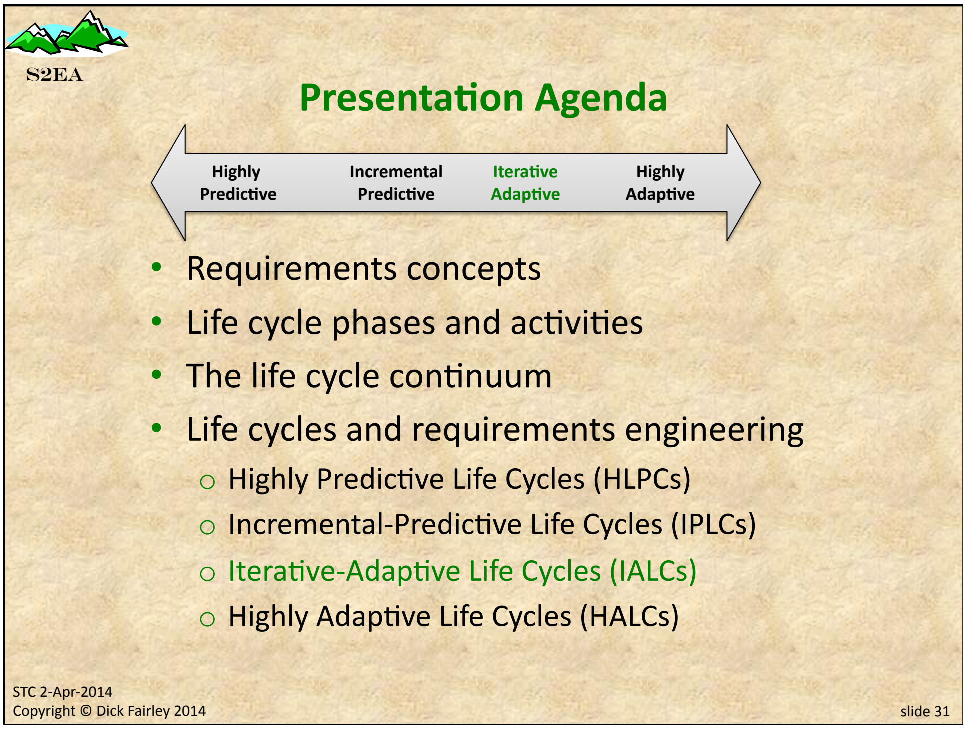
IPLCs can be successful when:

- Requirements can be mostly defined initially
- Some flexibility in requirements management will be permitted
  - plus dynamic allocation of resources and schedule to increments
  - or time-boxing of scheduled increments
- Periodic stakeholder involvement provides feedback
  - as a basis for adding, deleting, and revising requirements
- Early delivery of one or more subset capabilities is desired

STC 2-Apr-2014

Copyright © Dick Fairley 2014

slide 30



S2EA

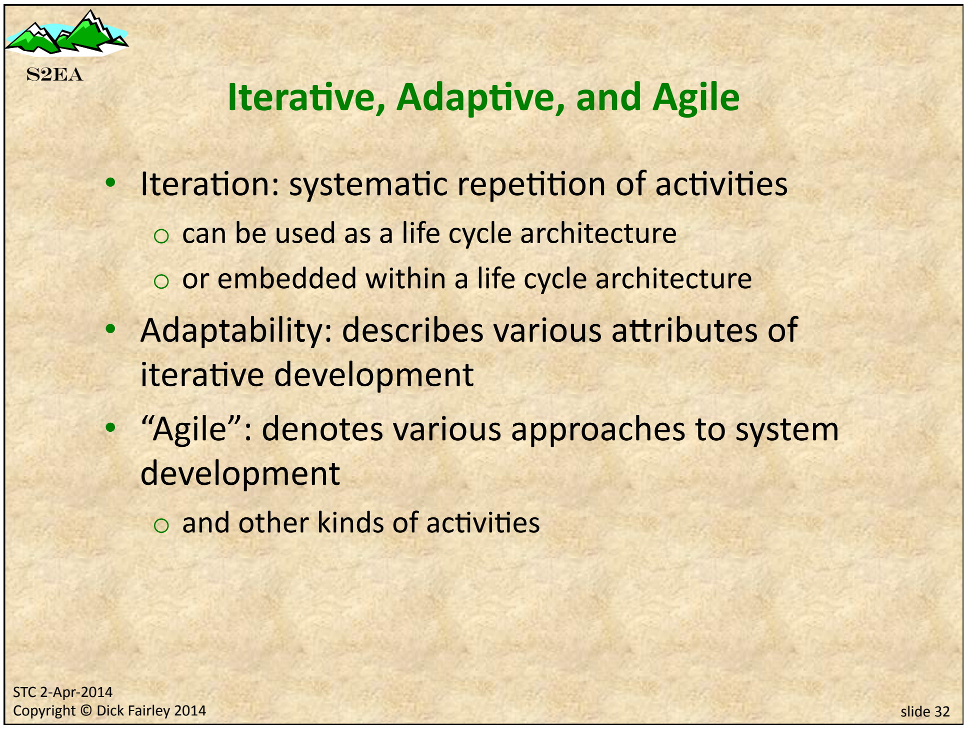
Presentation Agenda

Highly Predictive
Incremental Predictive
Iterative Adaptive
Highly Adaptive

- Requirements concepts
- Life cycle phases and activities
- The life cycle continuum
- Life cycles and requirements engineering
  - Highly Predictive Life Cycles (HLPCs)
  - Incremental-Predictive Life Cycles (IPLCs)
  - Iterative-Adaptive Life Cycles (IALCs)
  - Highly Adaptive Life Cycles (HALCs)

STC 2-Apr-2014

slide 31



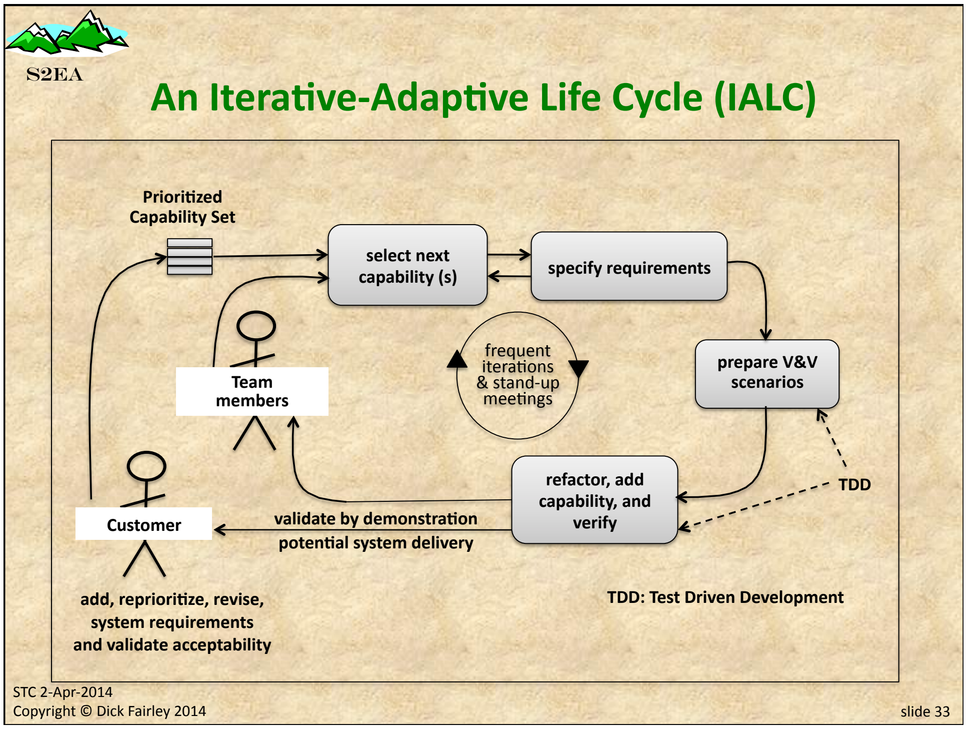
S2EA

Iterative, Adaptive, and Agile

- Iteration: systematic repetition of activities
  - can be used as a life cycle architecture
  - or embedded within a life cycle architecture
- Adaptability: describes various attributes of iterative development
- “Agile”: denotes various approaches to system development
  - and other kinds of activities

STC 2-Apr-2014

slide 32



S2EA

An Iterative-Adaptive Life Cycle (IALC)

Prioritized Capability Set

select next capability (s)

specify requirements

prepare V&V scenarios

frequent iterations & stand-up meetings

Team members

refactor, add capability, and verify

TDD

Customer

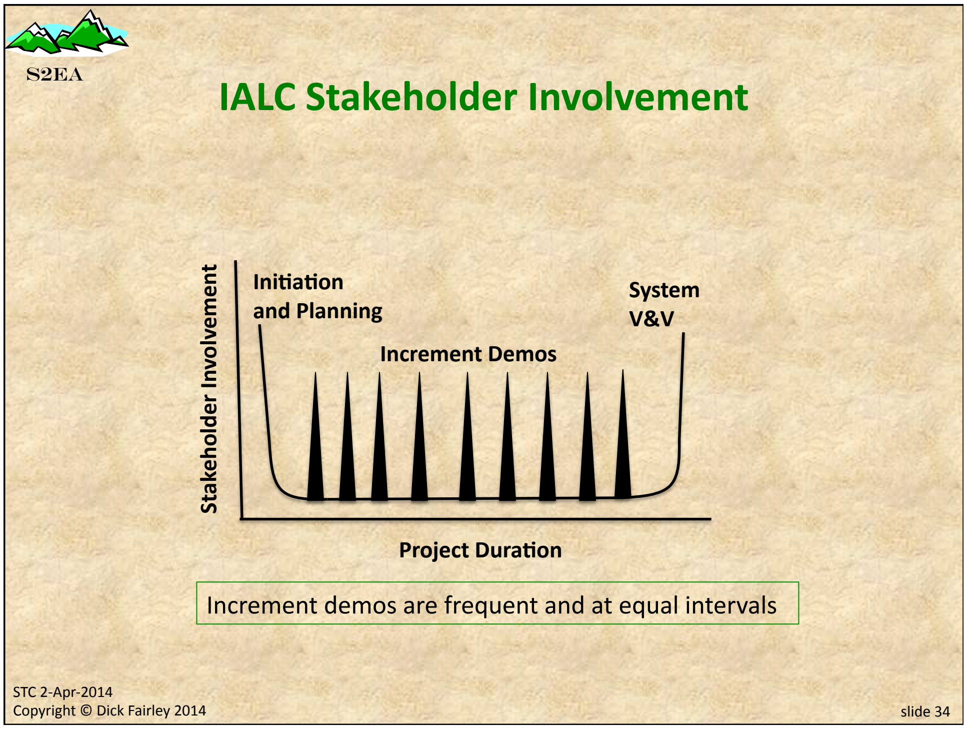
validate by demonstration potential system delivery

add, reprioritize, revise, system requirements and validate acceptability

TDD: Test Driven Development

STC 2-Apr-2014

slide 33



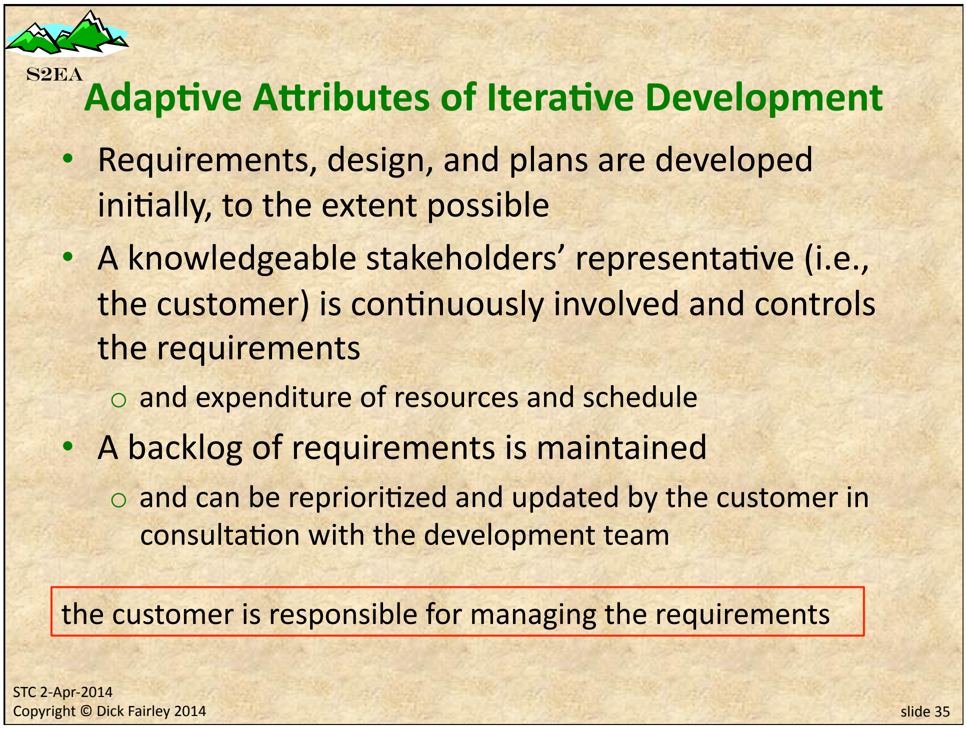
S2EA

IALC Stakeholder Involvement

Increment demos are frequent and at equal intervals

STC 2-Apr-2014

slide 34



S2EA

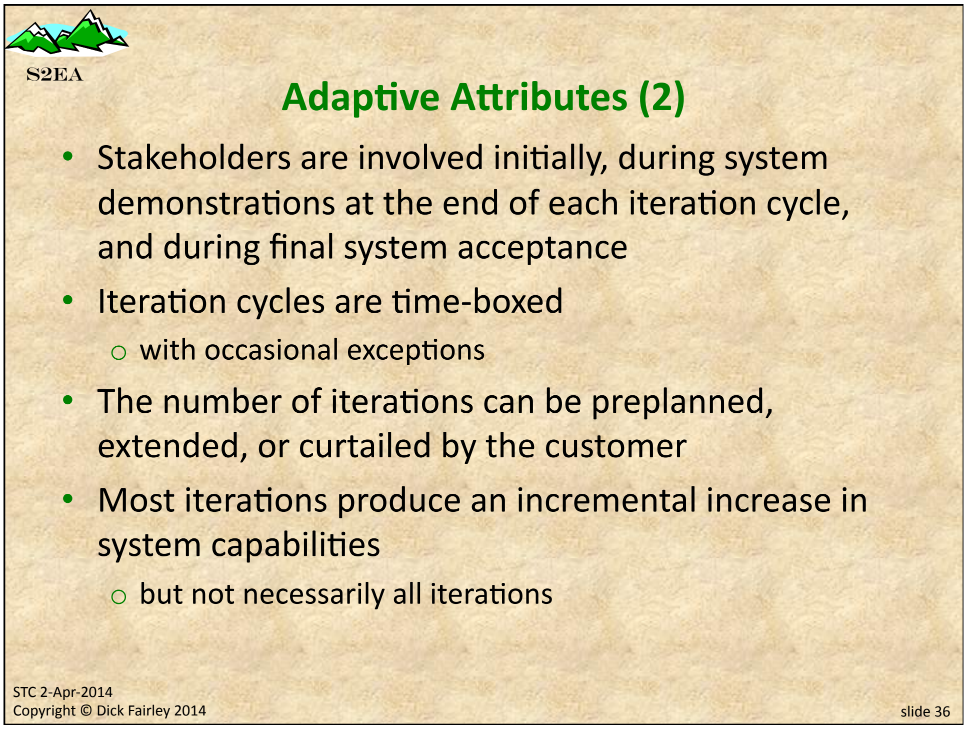
Adaptive Attributes of Iterative Development

- Requirements, design, and plans are developed initially, to the extent possible
- A knowledgeable stakeholders' representative (i.e., the customer) is continuously involved and controls the requirements
  - and expenditure of resources and schedule
- A backlog of requirements is maintained
  - and can be reprioritized and updated by the customer in consultation with the development team

the customer is responsible for managing the requirements

STC 2-Apr-2014

slide 35



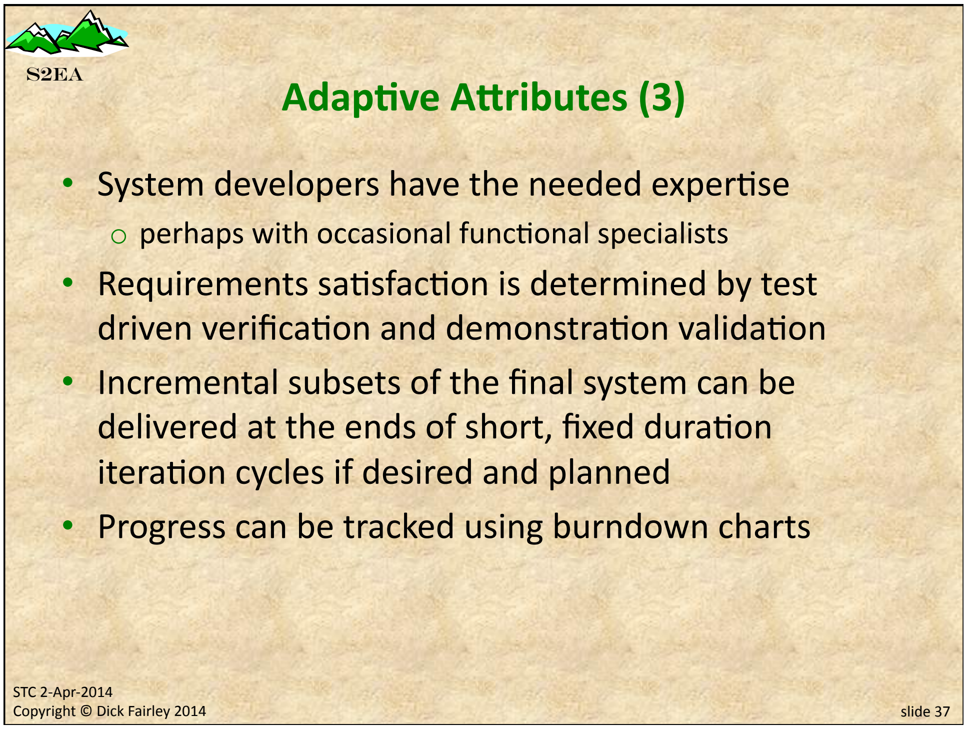
S2EA

Adaptive Attributes (2)

- Stakeholders are involved initially, during system demonstrations at the end of each iteration cycle, and during final system acceptance
- Iteration cycles are time-boxed
  - with occasional exceptions
- The number of iterations can be preplanned, extended, or curtailed by the customer
- Most iterations produce an incremental increase in system capabilities
  - but not necessarily all iterations

STC 2-Apr-2014

slide 36



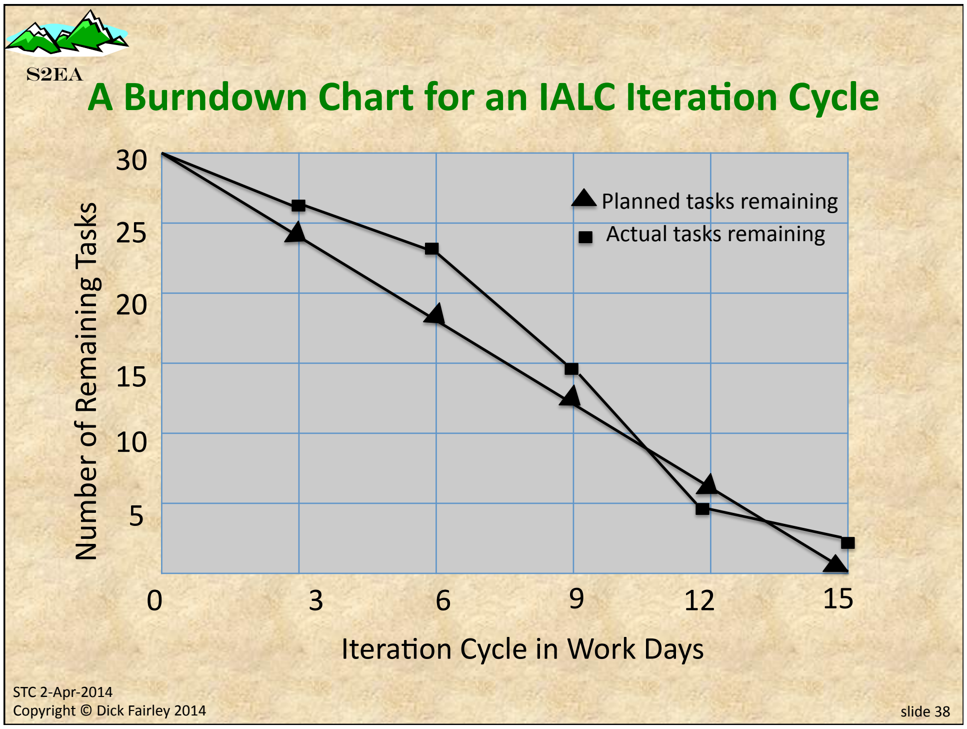
S2EA

Adaptive Attributes (3)

- System developers have the needed expertise
  - perhaps with occasional functional specialists
- Requirements satisfaction is determined by test driven verification and demonstration validation
- Incremental subsets of the final system can be delivered at the ends of short, fixed duration iteration cycles if desired and planned
- Progress can be tracked using burndown charts

STC 2-Apr-2014

slide 37

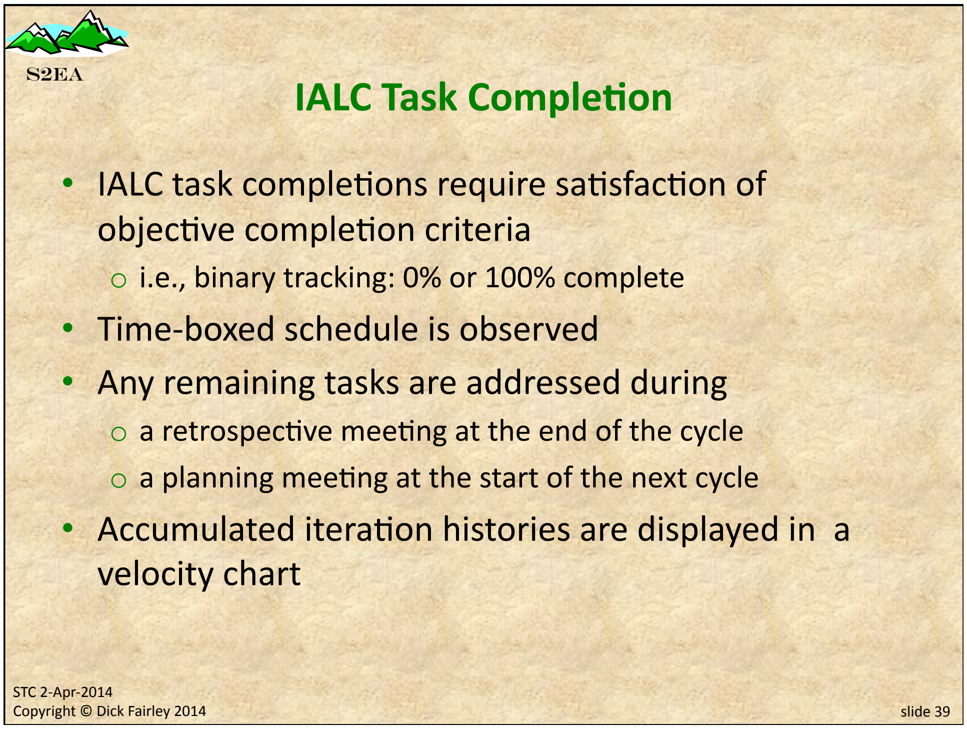


S2EA

A Burndown Chart for an IALC Iteration Cycle

STC 2-Apr-2014

slide 38



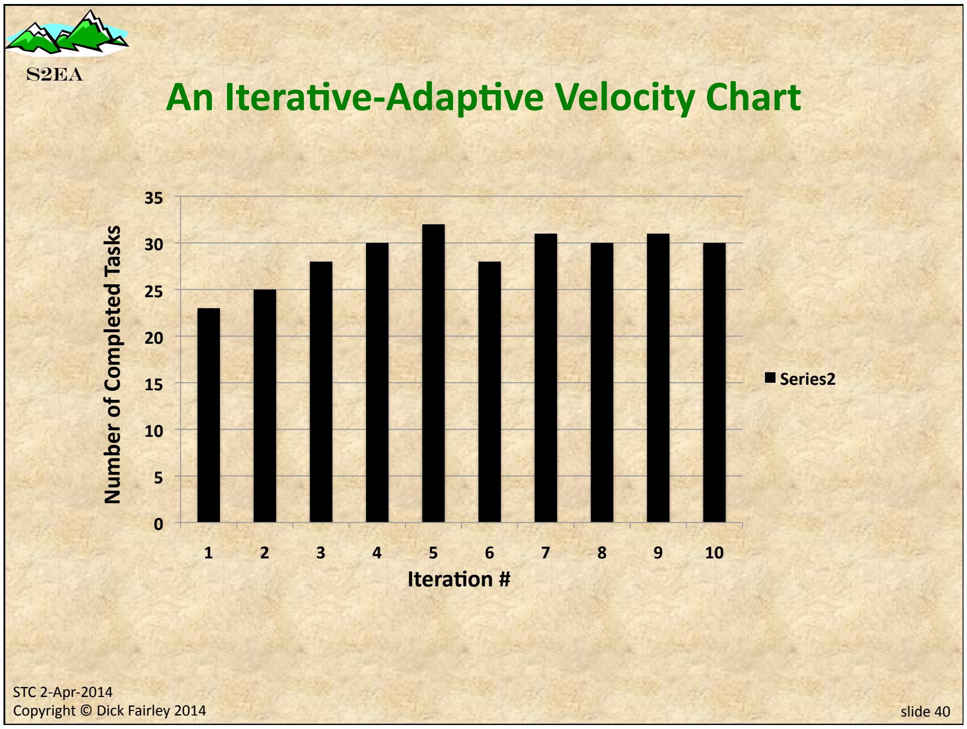
S2EA

IALC Task Completion

- IALC task completions require satisfaction of objective completion criteria
  - i.e., binary tracking: 0% or 100% complete
- Time-boxed schedule is observed
- Any remaining tasks are addressed during
  - a retrospective meeting at the end of the cycle
  - a planning meeting at the start of the next cycle
- Accumulated iteration histories are displayed in a velocity chart

STC 2-Apr-2014

slide 39

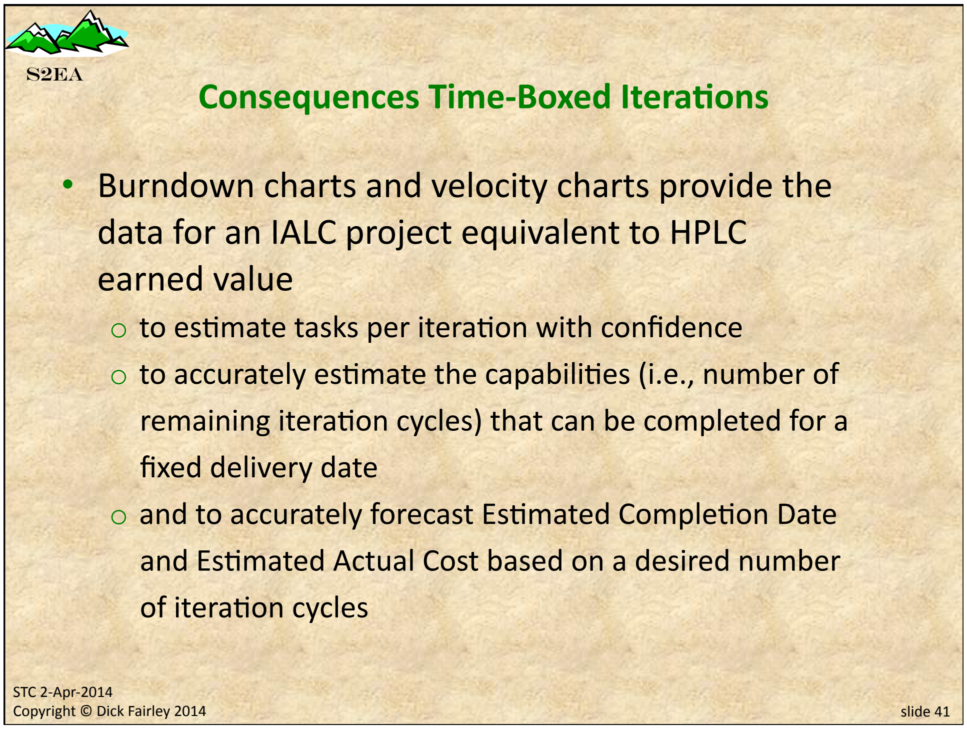


S2EA

An Iterative-Adaptive Velocity Chart

STC 2-Apr-2014

slide 40



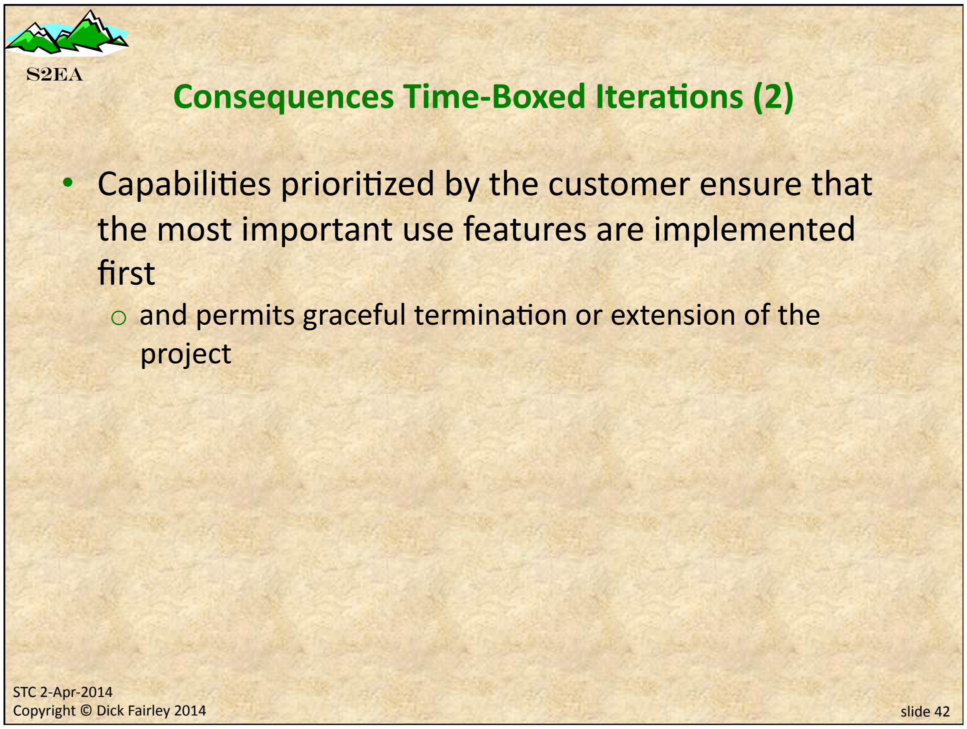
S2EA

Consequences Time-Boxed Iterations

- Burndown charts and velocity charts provide the data for an IALC project equivalent to HPLC earned value
  - ○ to estimate tasks per iteration with confidence
  - ○ to accurately estimate the capabilities (i.e., number of remaining iteration cycles) that can be completed for a fixed delivery date
  - ○ and to accurately forecast Estimated Completion Date and Estimated Actual Cost based on a desired number of iteration cycles

STC 2-Apr-2014

slide 41



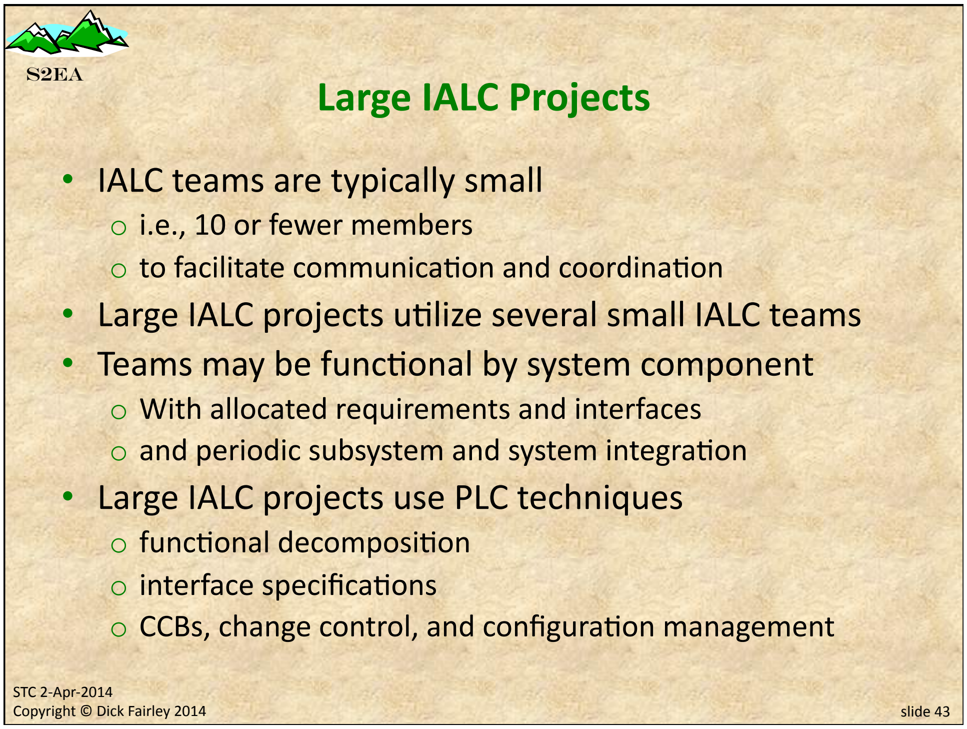
S2EA

Consequences Time-Boxed Iterations (2)

- Capabilities prioritized by the customer ensure that the most important use features are implemented first
  - ○ and permits graceful termination or extension of the project

STC 2-Apr-2014

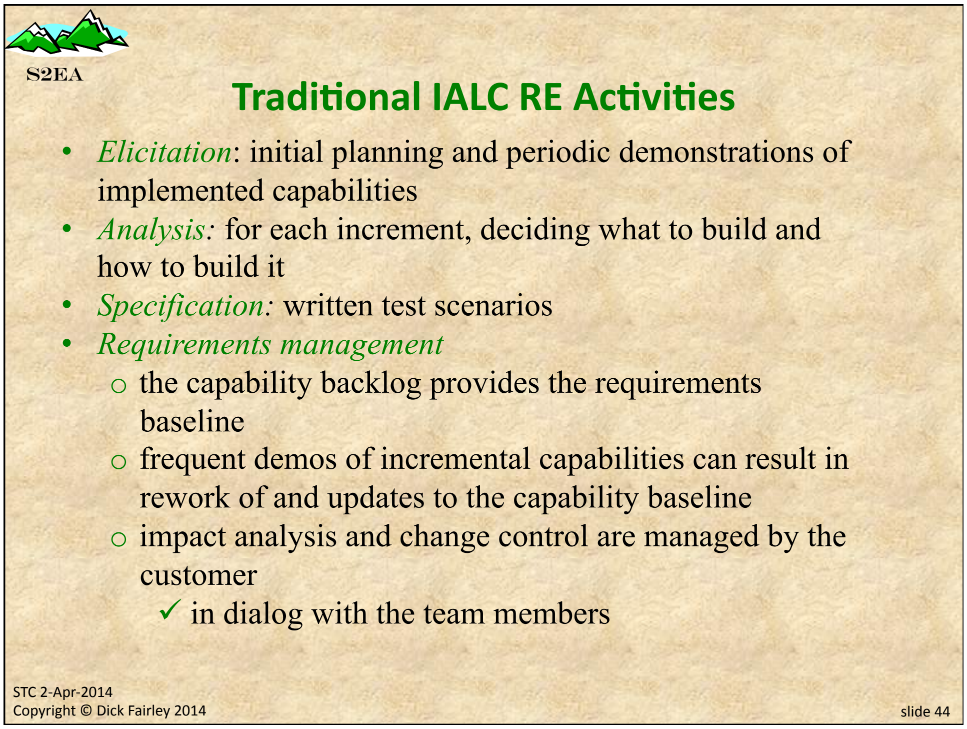
slide 42



S2EA

Large IALC Projects

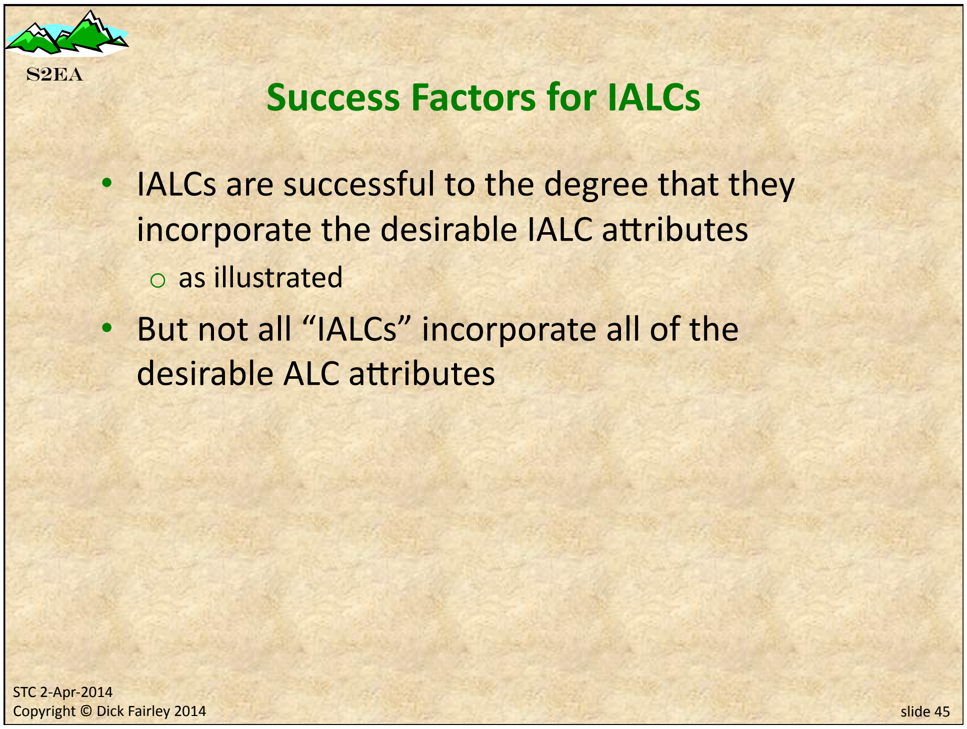
- IALC teams are typically small
  - i.e., 10 or fewer members
  - to facilitate communication and coordination
- Large IALC projects utilize several small IALC teams
- Teams may be functional by system component
  - With allocated requirements and interfaces
  - and periodic subsystem and system integration
- Large IALC projects use PLC techniques
  - functional decomposition
  - interface specifications
  - CCBs, change control, and configuration management



S2EA

Traditional IALC RE Activities

- *Elicitation*: initial planning and periodic demonstrations of implemented capabilities
- *Analysis:* for each increment, deciding what to build and how to build it
- *Specification:* written test scenarios
- *Requirements management*
  - the capability backlog provides the requirements baseline
  - frequent demos of incremental capabilities can result in rework of and updates to the capability baseline
  - impact analysis and change control are managed by the customer
    - ✓ in dialog with the team members



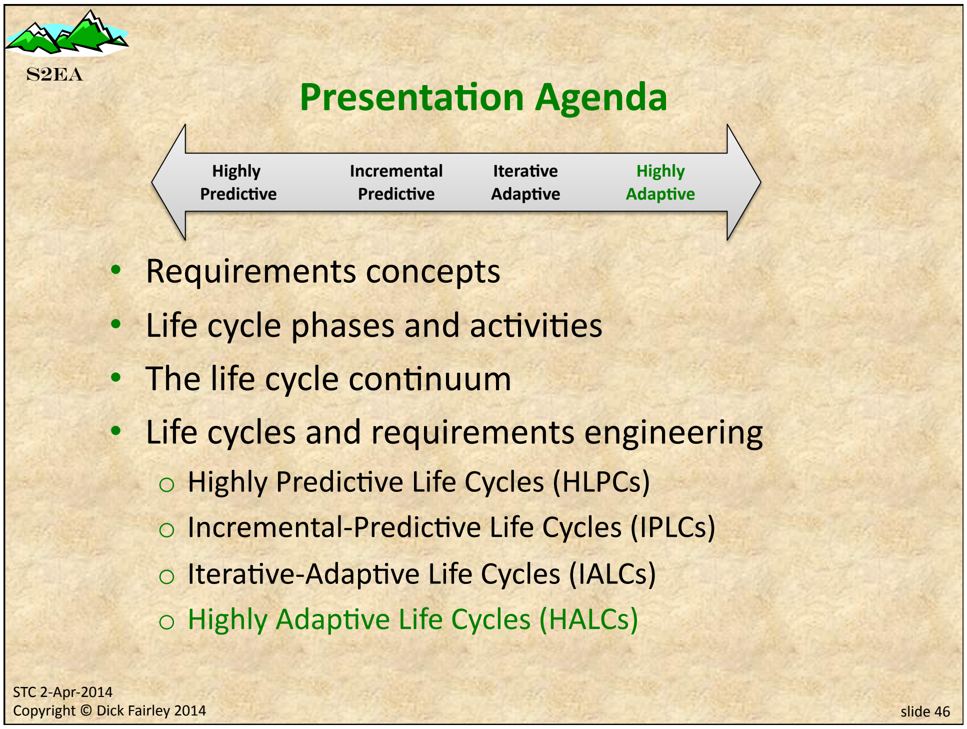
S2EA

Success Factors for IALCs

- IALCs are successful to the degree that they incorporate the desirable IALC attributes
  - ○ as illustrated
- But not all "IALCs" incorporate all of the desirable ALC attributes

STC 2-Apr-2014

slide 45



S2EA

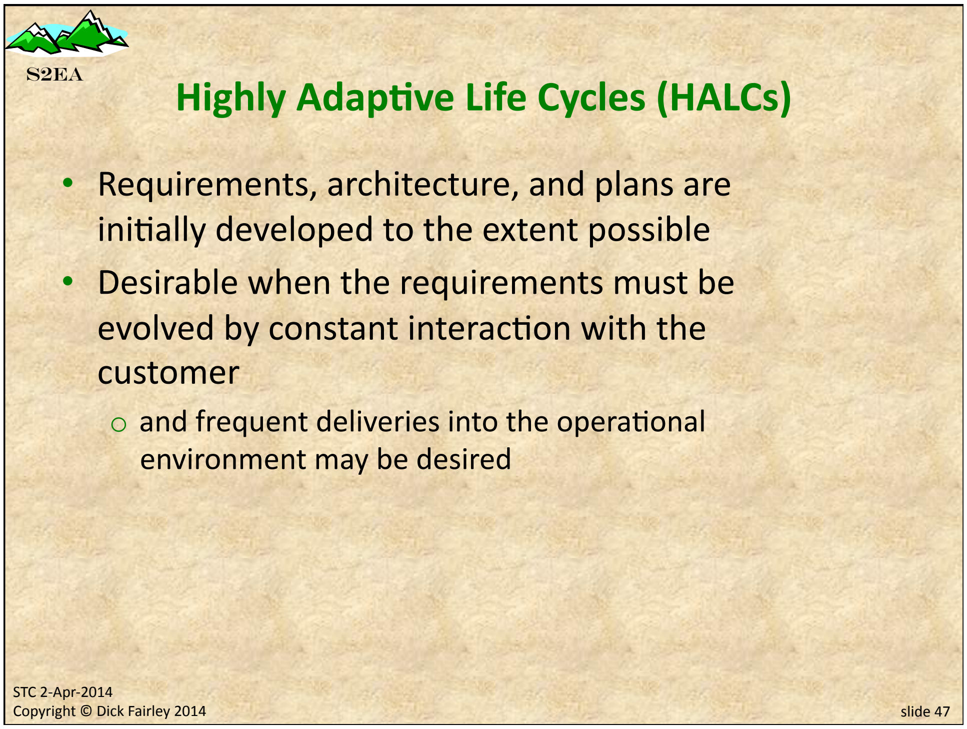
Presentation Agenda

Highly Predictive | Incremental Predictive | Iterative Adaptive | Highly Adaptive

- Requirements concepts
- Life cycle phases and activities
- The life cycle continuum
- Life cycles and requirements engineering
  - ○ Highly Predictive Life Cycles (HLPCs)
  - ○ Incremental-Predictive Life Cycles (IPLCs)
  - ○ Iterative-Adaptive Life Cycles (IALCs)
  - ○ Highly Adaptive Life Cycles (HALCs)

STC 2-Apr-2014

slide 46



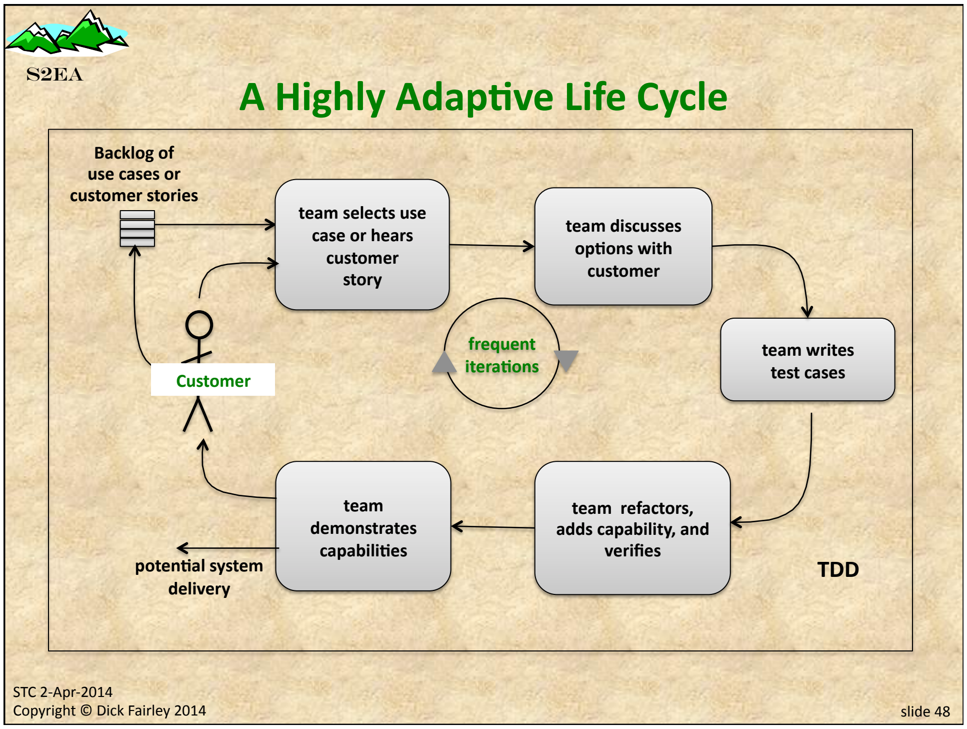
S2EA

Highly Adaptive Life Cycles (HALCs)

- Requirements, architecture, and plans are initially developed to the extent possible
- Desirable when the requirements must be evolved by constant interaction with the customer
  - ○ and frequent deliveries into the operational environment may be desired

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 47



S2EA

A Highly Adaptive Life Cycle

Backlog of use cases or customer stories

team selects use case or hears customer story

team discusses options with customer

team writes test cases

frequent iterations

Customer

team refactors, adds capability, and verifies

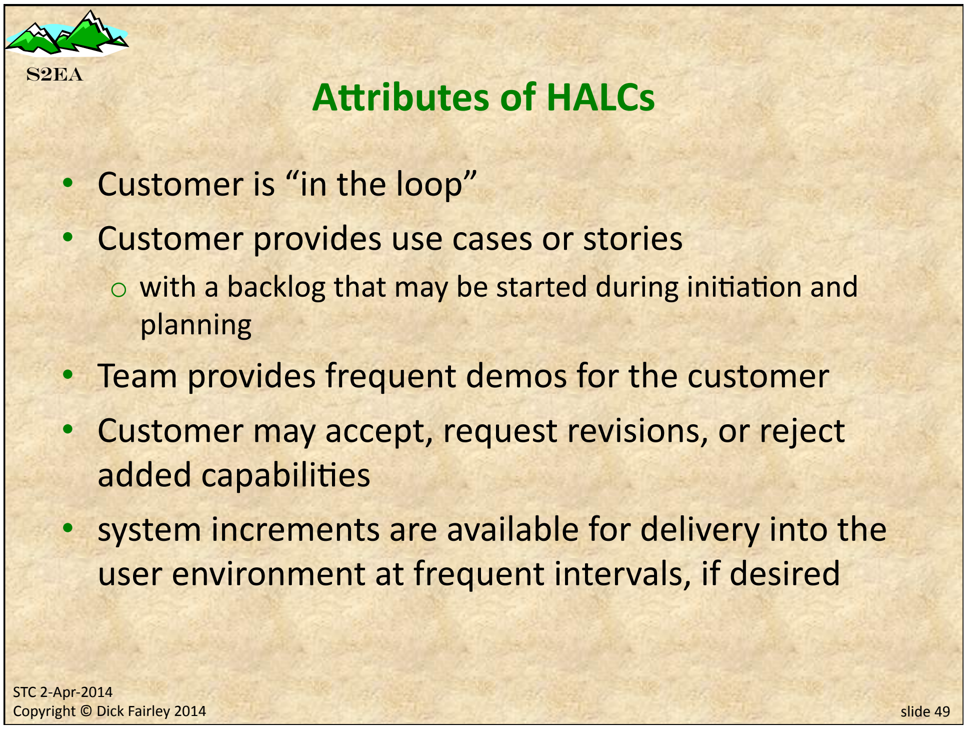
team demonstrates capabilities

potential system delivery

TDD

STC 2-Apr-2014
Copyright © Dick Fairley 2014

slide 48

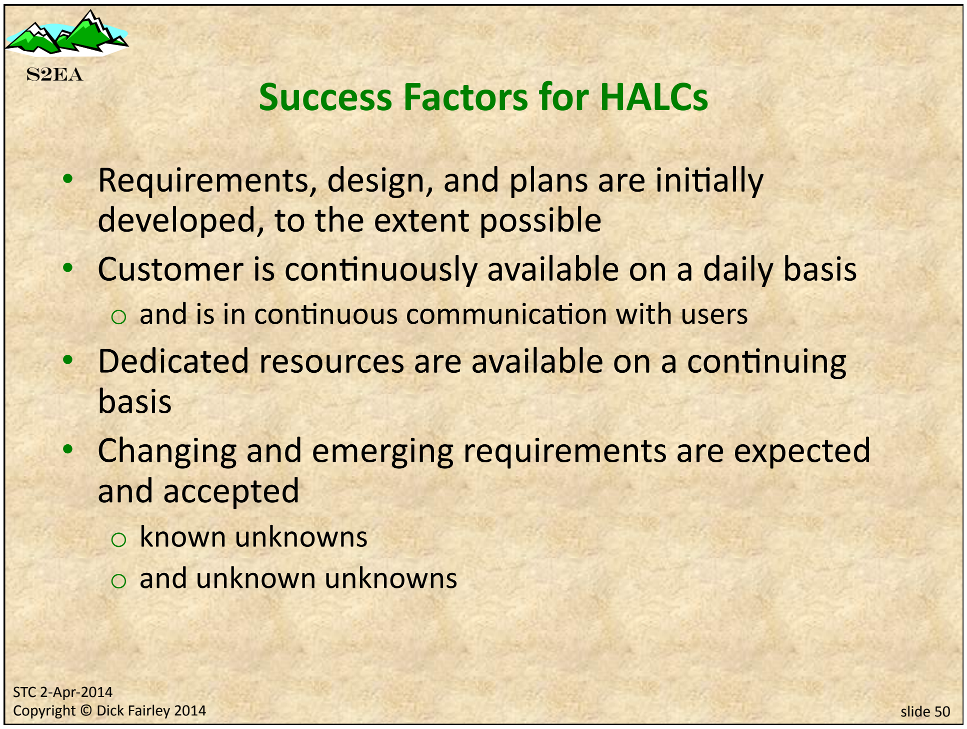


S2EA

Attributes of HALCs

- Customer is “in the loop”
- Customer provides use cases or stories
  - with a backlog that may be started during initiation and planning
- Team provides frequent demos for the customer
- Customer may accept, request revisions, or reject added capabilities
- system increments are available for delivery into the user environment at frequent intervals, if desired

STC 2-Apr-2014
Copyright © Dick Fairley 2014 slide 49

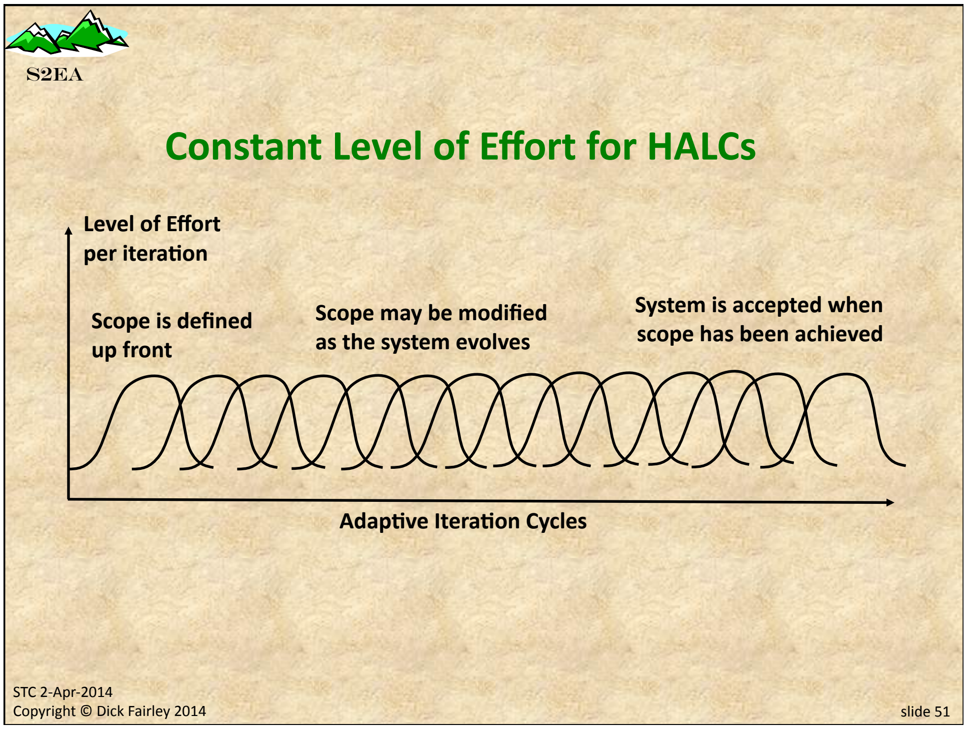


S2EA

Success Factors for HALCs

- Requirements, design, and plans are initially developed, to the extent possible
- Customer is continuously available on a daily basis
  - and is in continuous communication with users
- Dedicated resources are available on a continuing basis
- Changing and emerging requirements are expected and accepted
  - known unknowns
  - and unknown unknowns

STC 2-Apr-2014
Copyright © Dick Fairley 2014 slide 50

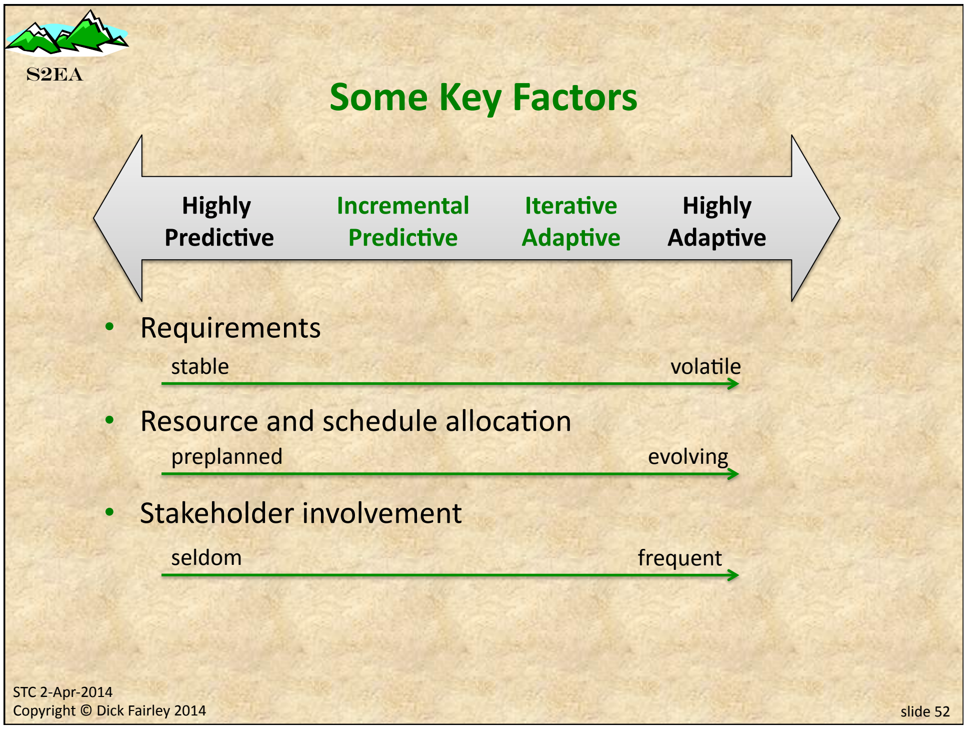


S2EA

Constant Level of Effort for HALCs

Level of Effort per iteration
Scope is defined up front
Scope may be modified as the system evolves
System is accepted when scope has been achieved
Adaptive Iteration Cycles

STC 2-Apr-2014
Copyright © Dick Fairley 2014
slide 51



S2EA

Some Key Factors

- Requirements

  stable → volatile

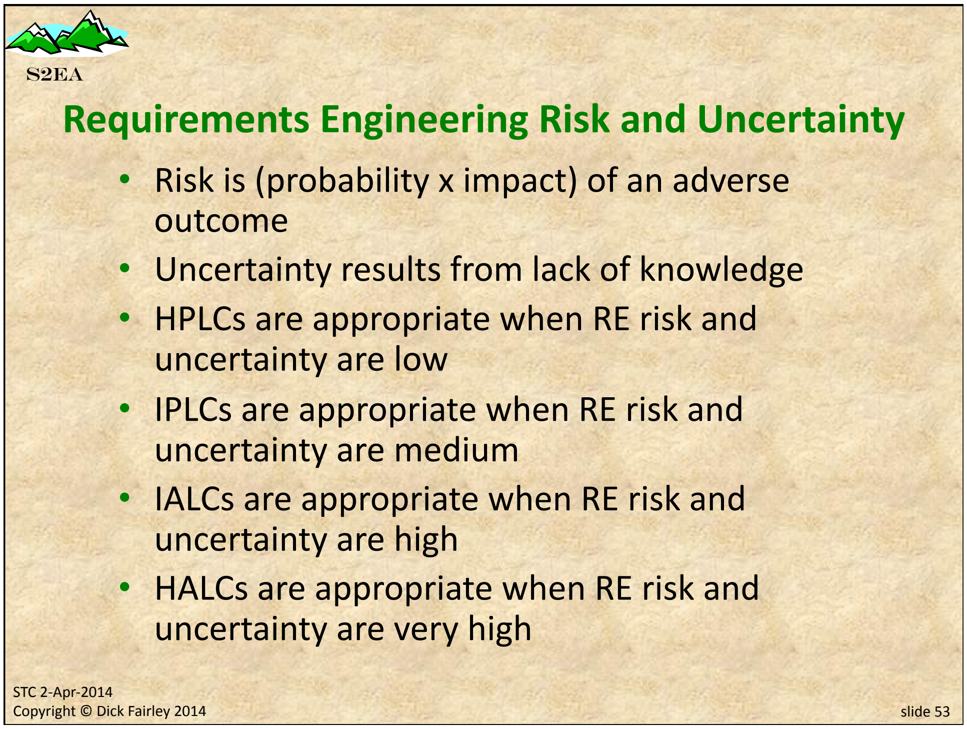
- Resource and schedule allocation

  preplanned → evolving

- Stakeholder involvement

  seldom → frequent

STC 2-Apr-2014
Copyright © Dick Fairley 2014
slide 52



S2EA

Requirements Engineering Risk and Uncertainty

- Risk is (probability x impact) of an adverse outcome
- Uncertainty results from lack of knowledge
- HPLCs are appropriate when RE risk and uncertainty are low
- IPLCs are appropriate when RE risk and uncertainty are medium
- IALCs are appropriate when RE risk and uncertainty are high
- HALCs are appropriate when RE risk and uncertainty are very high

STC 2-Apr-2014

slide 53