INCOSE Training Webinar
System Implementation

Developed and presented by

Richard E. (Dick) Fairley

dickfairley@gmail.com

Brief Bio

dickfairley@gmail.com

Dick Fairley is a long-time member of INCOSE. He is an author of the Guide to the Systems Engineering Body of Knowledge (SEBoK V 1.0), an editor of the present SEBoK V 1.9, an INCOSE commissioner for ABET accreditation of systems engineering degree programs, and INCOSE liaison to the IEEE Systems Council.

He is principal associate of Systems and Software Engineering Associates (S2EA), a consulting and training company. Dick is a former professor and associate dean of the Oregon Graduate Institute and past Chair of the IEEE Computer Society Systems and Software Engineering Committee. He was co-editor of the Software Engineering Body of Knowledge (SWEBOK V 1.3) and leader of the teams that developed the Software Engineering Competency Model (SWECOM) and the Software Extension to the PMI Guide to the Project Management Body of Knowledge (SWX).

His Bachelors and Masters degrees are in electrical engineering and his PhD is in computer science and applied math. He worked in industry as an electrical and systems engineer before returning to school to obtain his PhD from UCLA.

Dick and his wife Mary Jane live in the Colorado mountains northwest of Colorado Springs. He enjoys listening to jazz, hiking, and skiing. They enjoy motorcycling together.

Five training webinars

1. Stakeholders' requirements
   November 8, 2018
2. System requirements
   November 15, 2018
3. System architecture
   January 3, 2019
4. System design
   January 10, 2019
5. System implementation
   January 17, 2019

All 5 webinars will be presented at noon EST and recorded for later viewing

Accessing the training webinars

- These and other INCOSE webinars are recorded for listening and downloading of the presentation slides
- First, sign on to:

  https://connect.incose.org/Library/Tutorials/training/SitePages/Home.aspx

- Then scroll to the bottom of the page. Under Systems Engineering Technical Processes you will see the technical process training webinars

For assistance, contact John Clark or Gabriela Coe at john.clark@incose.org or gabriela.coe@incose.org

Primary references for this webinar

Clause 6.4.5 of ISO/EIC/IEEE Standards 15288:2015 and 12207:2017

- **15288:** Systems and Software Engineering - - System life cycle processes

  https://www.iso.org/standard/63711.html
  https://standards.ieee.org/standard/15288-2015.html

and

- **12207:** Systems and Software Engineering - Software Life Cycle Processes

  https://www.iso.org/standard/63712.html
  https://standards.ieee.org/standard/12207-2017.html

NOTE

- Clause 6.4 of 15288 and 12207 (Technical Processes) provides the framework for these training webinars
- Clause 6.4 presents 14 technical processes of systems engineering as clauses 6.4.1 through 6.4.14, in sequence
- But different approaches to system development typically use the technical processes in various iterated, overlapped, and repeated ways

  and concurrently and recursively as required

> Listing of the processes in sequence does not imply they should be accomplished in linear order

Note

The Guide to the Systems Engineering Body of Knowledge (SEBoK Part 4) covers the following applications of systems engineering:

- Product Systems Engineering
- Service Systems Engineering
- Enterprise Systems Engineering
- Systems of Systems (SoS)
- Healthcare Systems Engineering

https://www.sebokwiki.org/wiki/Applications_of_Systems_Engineering

Another primary reference

Systems Engineering for Software-enabled Physical Systems*

Physical systems: embedded, IoT, cyber-physical, C2 systems, transportation, health care, and others

To be published by Wiley in May 2019

References for this webinar:

Chapter 4: Process models for system development

Chapter 5: The integrated iterative-incremental system development model

Chapter 9: System implementation and delivery

Other reference materials

1. INCOSE Systems Engineering Handbook
2. INCOSE Systems Engineering Competency Framework
3. The Guide to the Systems Engineering Body of Knowledge

   https://sebokwiki.org

- The Systems Engineering Handbook, Competency Framework, and other materials are available for free downloading from the INCOSE Store:

  https://connect.incose.org/pages/store.aspx
- SEBoK is universally accessible at no cost

Note

- Access to these references is not necessary for these training webinars
  - However, your organization or library may have access if you desire to see them

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

A brief review

Webinar 1:

Business or Mission Analysis: 15288 Clause 6.4.1

business need, mission need, or opportunity

Stakeholder Needs and Requirements Definition: Clause 6.4.2

identify stakeholders, categorize and prioritize their requirements, iv&v wrt need or opportunity

Webinar 2: System Requirements Definition: Clause 6.4.3

categorize, prioritize, and iv&v wrt stakeholders' requirements

Webinar 3: System Architecture Definition: Clause 6.4.4

system structure and behavior; SysML diagrams, iv&v

Webinar 4: System Design Definition: Clause 6.4.5

System definition, system design methods and techniques, iv&v

Five training webinars

Webinar 1:

Business or Mission Analysis: 15288 Clause 6.4.1

Stakeholder Needs and Requirements Definition: Clause 6.4.2

Webinar 2: System Requirements Definition: Clause 6.4.3

Webinar 3: System Architecture Definition: Clause 6.4.4

Webinar 4: System Design Definition: Clause 6.4.5

Webinar 5: System Implementation: Clauses 6.4.7 and 6.4.8

15288 Clause 6.4.6 is the System Analysis process

Clause 6.4.6 System analysis process

6.4.6.1 Purpose

"The purpose of the System Analysis process is to provide a rigorous basis of data and information for technical understanding to aid decision-making across the life cycle."

- "The System Analysis process can be used to develop the inputs needed for any technical assessment. It can provide confidence in the utility and integrity of system requirements, architecture, and design."
- "System analysis can include a wide range of differing techniques. Formality and rigor of the analysis will depend on the criticality of the information need or work product supported, the amount of information/ data available, the size of the project, and the schedule for the results."
- "NOTE This process is often used in conjunction with the Decision Management process (Clause 6.3.3)"

Clause 6.4.6 was added to 15288:2015

Note

The System Analysis process can be applied anytime (and repeatedly) during system development but system analysis is especially important prior to committing to system implementation

The System Analysis process was added to 15288:2015

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

System implementation processes

System implementation consists of two 15288 processes: implementing system elements and integrating them

1. 15288 Clause 6.4.7 Implementation process

   6.4.7.1 Purpose: "The purpose of the Implementation process is to realize a specified system element."

2. 15288 Clause 6.4.8 Integration process

   6.4.8.1 Purpose: "The purpose of the Integration process is to synthesize a set of system elements into a realized system (product or service) that satisfies system requirements, architecture, and design."

"to realize" is to make real

15288 Clause 4.1 Terms and definitions

Clause 4.1.47 system element

- "A system element is a discrete part of a system that can be implemented to fulfill specified requirements."
- "Kinds of system elements include hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities or any combination."

> A system element is a self-contained unit that has allocated requirements and interfaces that can be implemented separate from other system elements

System elements

A system element:

- can be large or small
- May be unidisciplinary or multidisciplinary
- May be system-critical or system-enabling
- Can be independently verified and validated
- May require a test environment for realization and iv&v
    - The test environment may include simulated elements, previously verified system elements, and real and simulated interfaces to other system elements and to the system environment
- May be a configuration item or part of a larger configuration item

15288 Clause 6.4.7.3 Perform implementation

Terminology

6.4.7.3 b) 1) Note

"i) Hardware elements are either acquired or *fabricated*.

ii) Software elements are either acquired or *developed* . . . ISO/IEC/IEEE 12207 applies to system elements realized in software."

Terminology: software elements are developed or *constructed*

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

15288 Clause 6.4.8 Integration process

Clause 6.4.8.1: Purpose

"The purpose of the integration process is to assemble a set of system elements into a realized system."

"Also, interfaces are identified and activated to enable interoperation of the system elements as intended."

Clause 6.4.8.3 Note 1 indicates that the Verification process is used to verify that interfaces satisfy their requirements

The Integration process focuses on verifying the interfaces among previously verified and validated system elements

15288 Clause 6.4.8 Integration process (2)

Clause 6.4.8 Notes

The notes also indicate that integration is performed at all levels of the system hierarchy to iteratively combine partial or complete system configurations

Integration is applied recursively for successive levels of the system hierarchy

And can be applied to partially completed systems

15288 Clause 6.4.8.3 Activities and tasks (2)

6.4.8.3.a) 2) Define the integration strategy

- NOTE 1 indicates that the integration strategy is predefined to specify a sequenced order for assembling the implemented system elements
- NOTE 2 indicates that the integration strategy often provides for verification of progressively more complete configurations of system elements

NOTES 1 and 2 accommodate iterative, incremental, and other forms of system development

Note

The phrase

“progressively more complete configurations of system elements”

indicates the importance of the Configuration Management process for the Integration process

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

A small familiar example

- A stakeholders' requirement:

  The customers of our financial institution should be able conduct financial transactions at convenient times and in convenient locations; safely and securely

- Some ATM system capabilities:

  C1: ability to provide secure customer authentication

  C2: ability to provide secure session termination

  C3: ability to check account balances

  C4: ability to withdraw funds

  C5: ability to deposit funds into accounts

  C6: ability to transfer funds between accounts

  C7: ability to handle exception conditions that may arise

Physical security of customers will be addressed in the operational requirements

Implementing ATM capabilities

Capabilities	ATM Hardware, Software, and Interfaces						
	Card Reader	Display Screen	Keypad	Cash Safe	Cash Dispenser	Funds Depository	Printer
Customer authentication	X	X	X				
Session termination		X	X				X
Balance query		X	X				
Cash withdrawal		X	X	X	X		
Funds deposit		X	X			X	
Funds transfer		X	X				

Note: designing and implementing the card reader, display screen, and keypad includes designing and implementing the user interface

ATM hardware, software, and interfaces

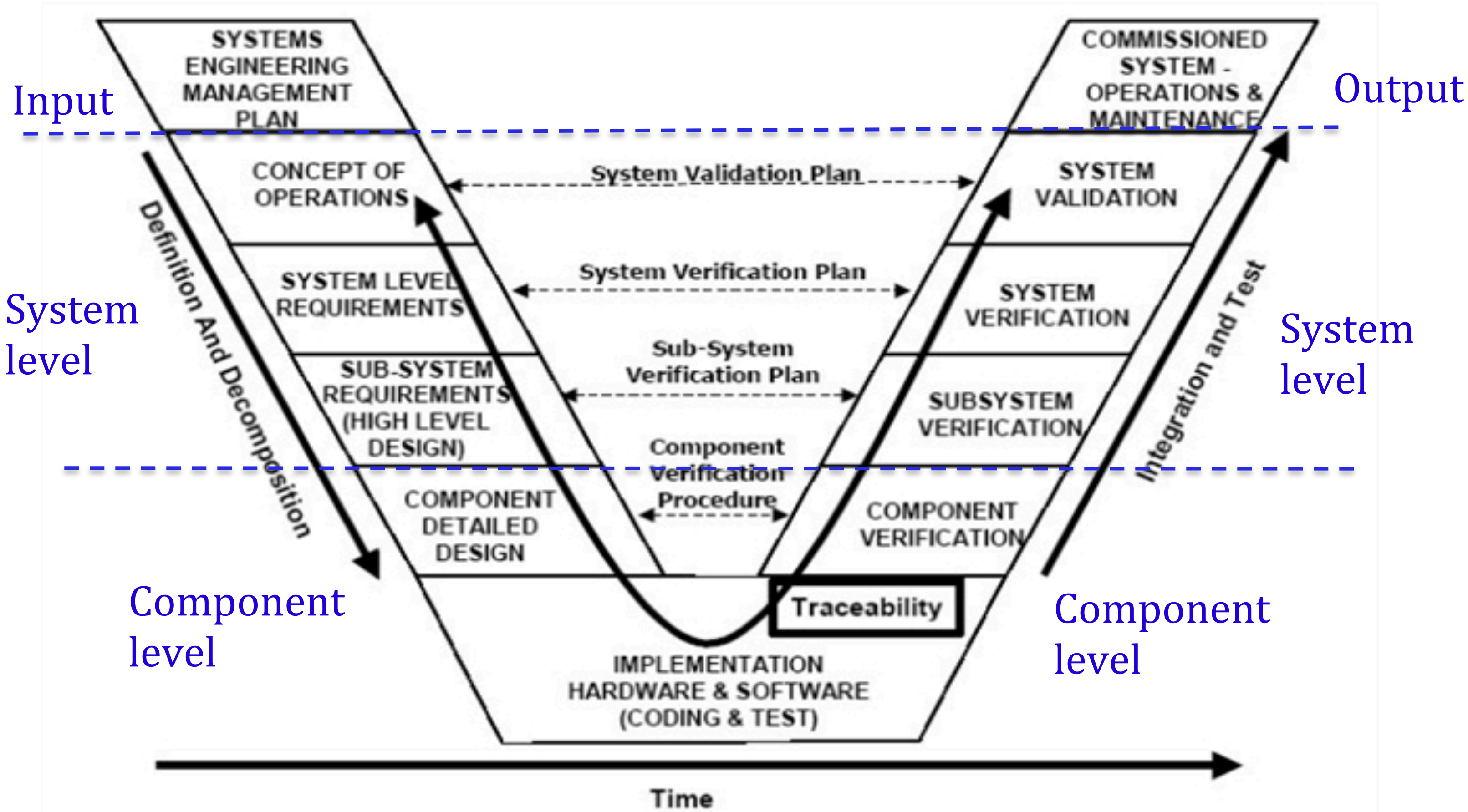
Hardware to be procured and fabricated:

- Card reader, keypad, display screen, cash safe, cash dispenser, funds depositor, and printer, PLUS
- the service technician's interface panel, physical security sensors and alarms, power supply, cabling, housing and mounting brackets, various connectors (among the internal hardware elements, the data interface connector to the financial institution, the power supply, and sensors and alarms)

Software to be procured and developed (constructed)

- Functionality, behavior, performance characteristics for each system element
  - Software elements and software-enabled physical elements
- Data and control interfaces among system elements and to the environment
- Exception detection and error handling

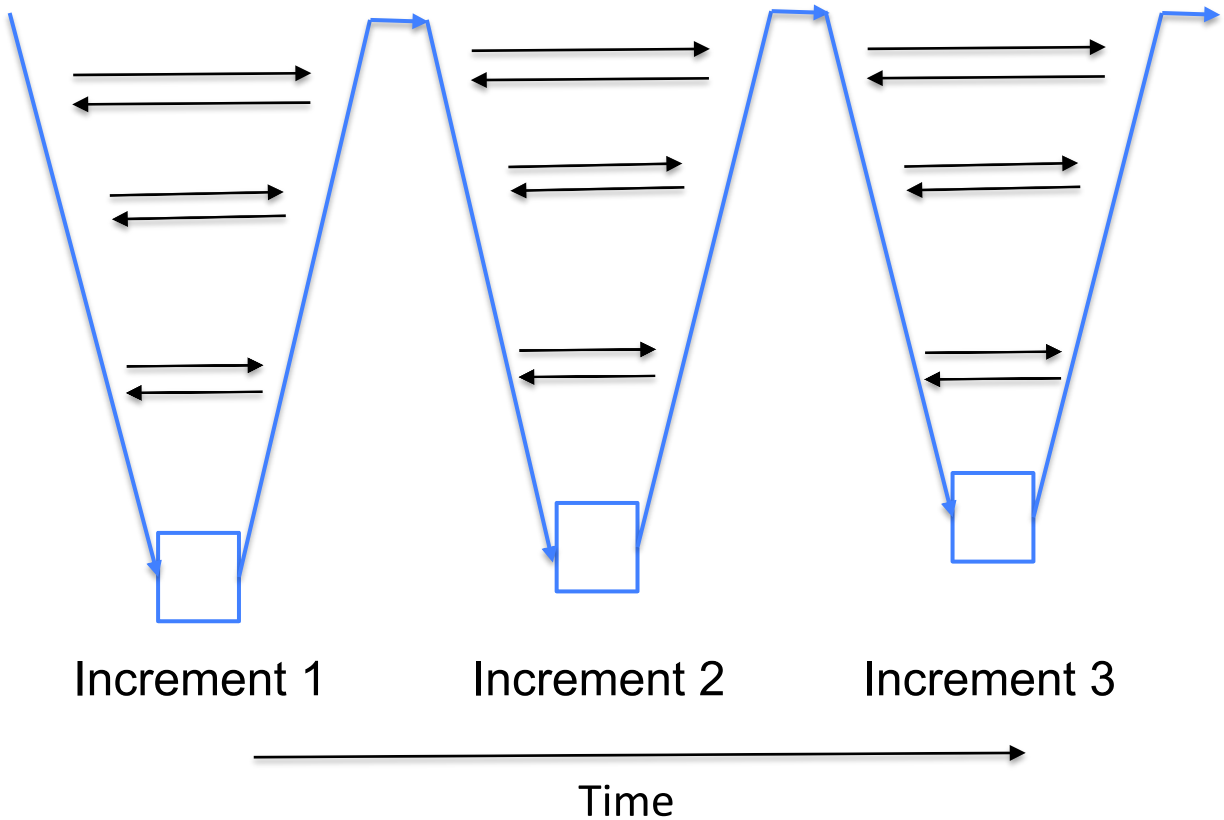
A Vee model for system development*



Iteration and revision are not illustrated

*http://www.engineering.com/portals/0/BlogFiles/Systems_Engineering_V_diagram.jpg

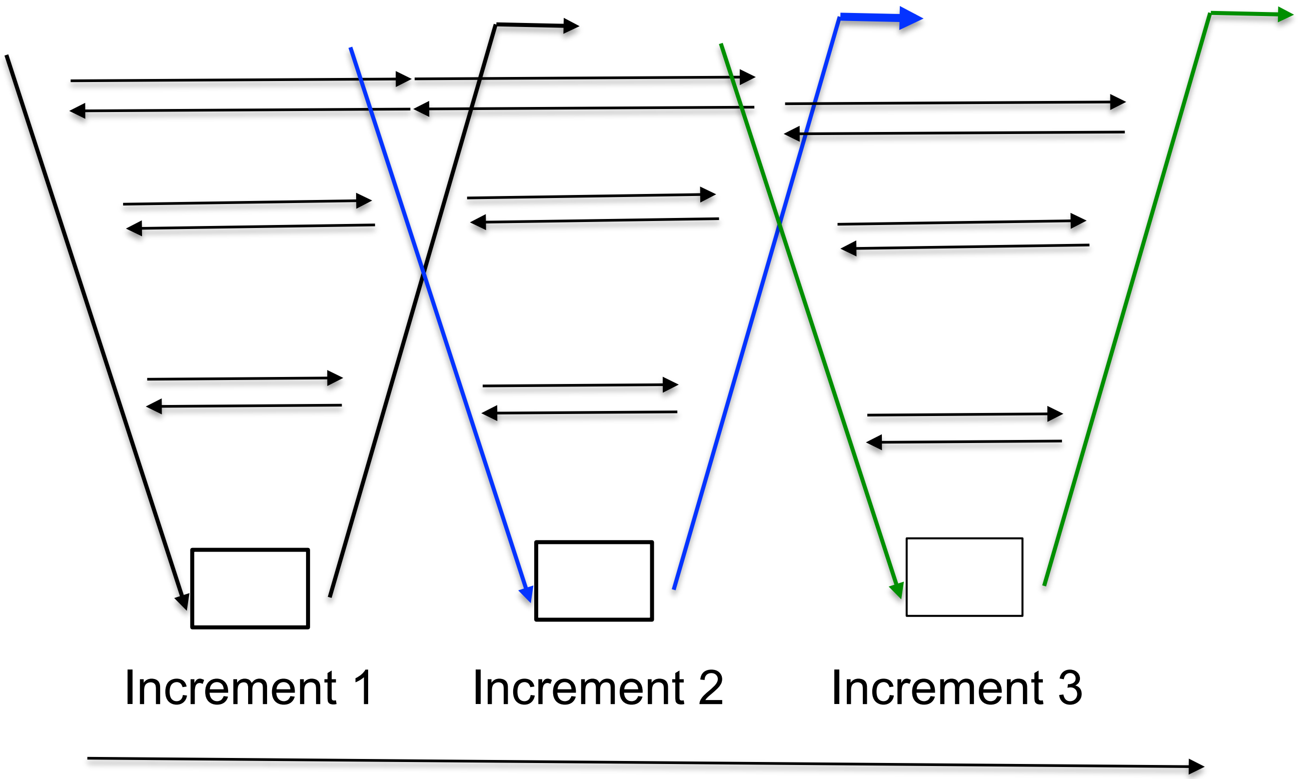
An incremental Vee system development model
Source: Figure 4.8*



Each increment incorporates the results of all previous increments

* Systems Engineering for Software-enabled Physical Systems; R. Fairley, Wiley 2019

An overlapping incremental Vee development model
Source: Figure 4.9*



* Systems Engineering for Software-enabled Physical Systems; R. Fairley, Wiley 2019

Incremental Vees and overlapping Vees

Incremental Vees

- Each Vee produces a demonstrable system increment
  - Can provide periodic early warning of problems
  - Can provide the platform for the next increment
  - And may include simulated black box elements

Overlapping Vees

- Overlapped incremental Vees can shorten the schedule
  - Provided sufficient resources are available
  - Caution must be taken to not get too far ahead on the next increment while completing the current increment

ATM incremental hardware-first implementation

Implementation priorities:

- Increment 1: Secure Customer Authentication capability
- Increment 2: Secure Session Termination capability
- Increment 3: Balance Query capability
- Increment 4: Cash Withdrawal capability
- Increment 5: Funds Deposit capability
- Increment 6: Funds Transfer capability

Note 1: capabilities developed first are tested most

Note 2: capabilities developed first are used by subsequently implemented capabilities

Note 3: durations for developing different increments may be different

ATM hardware, software, and interfaces

Hardware to be procured and fabricated:

- Card reader, keypad, display screen, cash safe, cash dispenser, funds depositor, and printer, PLUS
- the service technician's interface panel, physical security sensors and alarms, power supply, cabling, housing and mounting brackets, various connectors (among the internal hardware elements, the data interface connector to the financial institution, the power supply, and sensors and alarms)

Software to be procured and developed (i.e., constructed)

- Functionality, behavior, performance characteristics for each system element
  - Software elements and software-enabled physical elements
- Data and control interfaces among system elements and to the environment
- Exception detection and error handling

ATM iterative software-first implementation

Implementation priorities:

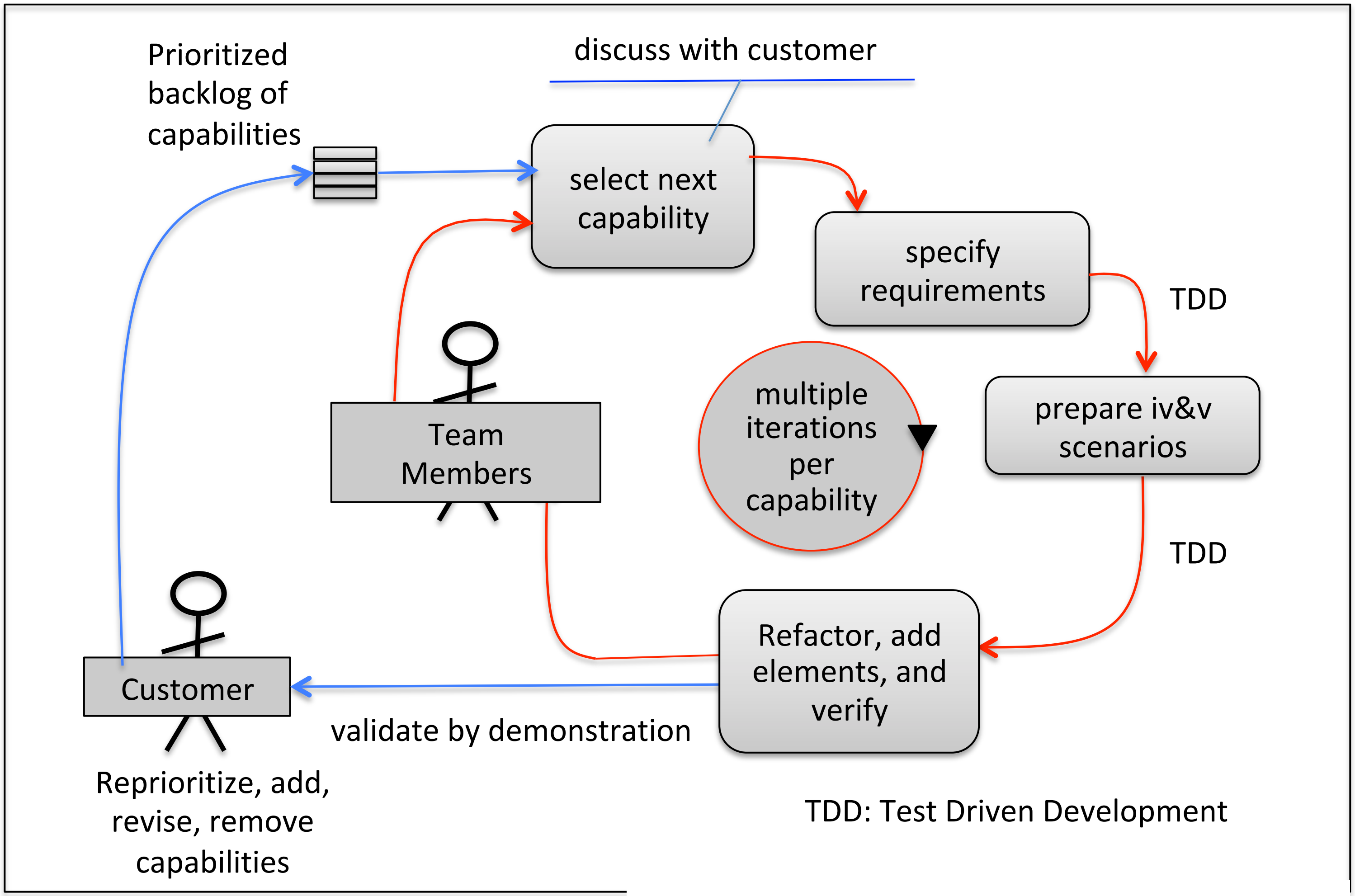
- Capability 1: Secure Customer Authentication capability
- Capability 2: Secure Session Termination capability
- Capability 3: Balance Query capability
- Capability 4: Cash Withdrawal capability
- Capability 5: Funds Deposit capability
- Capability 6: Funds Transfer capability

Note 1: capabilities developed first are tested most

Note 2: capabilities developed first are used by subsequently implemented capabilities

Note 3: durations for developing different capabilities may be different

Iterative software construction
Source: Figure 4.10*



* Systems Engineering for Software-enabled Physical Systems; R. Fairley, Wiley 2019

A big problem

- How to coordinate development of hardware, software, and interfaces?
  - Hardware-first?
  - Software-first?
  - Concurrent development?

Development of hardware and software

- Hardware-first and software-first development methods result in interface defects, rework, and late system delivery
- Hardware-first: the software engineers are waiting for their requirements, detailed interface specifications, and design constraints – which probably will not provide the details they need
- Software-first: the hardware engineers are waiting for their requirements, detailed interface specifications, and design constraints – which probably will not provide the details they need

Uncoordinated concurrent development is not the solution

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

MBSE*

- Model-based systems engineering (MBSE) uses a formalized modeling approach to develop visual models beginning in the requirements phase and continuing through the development phases and later life cycle phases.
  - SysML, other notations, and supporting tools are used
- Models can supplement and replace documents as a medium of communication among engineers
  - And can facilitate informal and formal verification and validation

* INCOSE SE Vision 2020 (INCOSE-TP-2004-004-02, Sep 2007)

MBSE is a major initiative of INCOSE and many system development organizations

M&SBSE

- Modeling and simulation-based systems engineering (M&SBSE) animates execution of MBSE models of system requirements, architecture, design, and implementation
- Methods and tools for M&SBSE are evolving rapidly

  But are still immature

Agenda for System Implementation

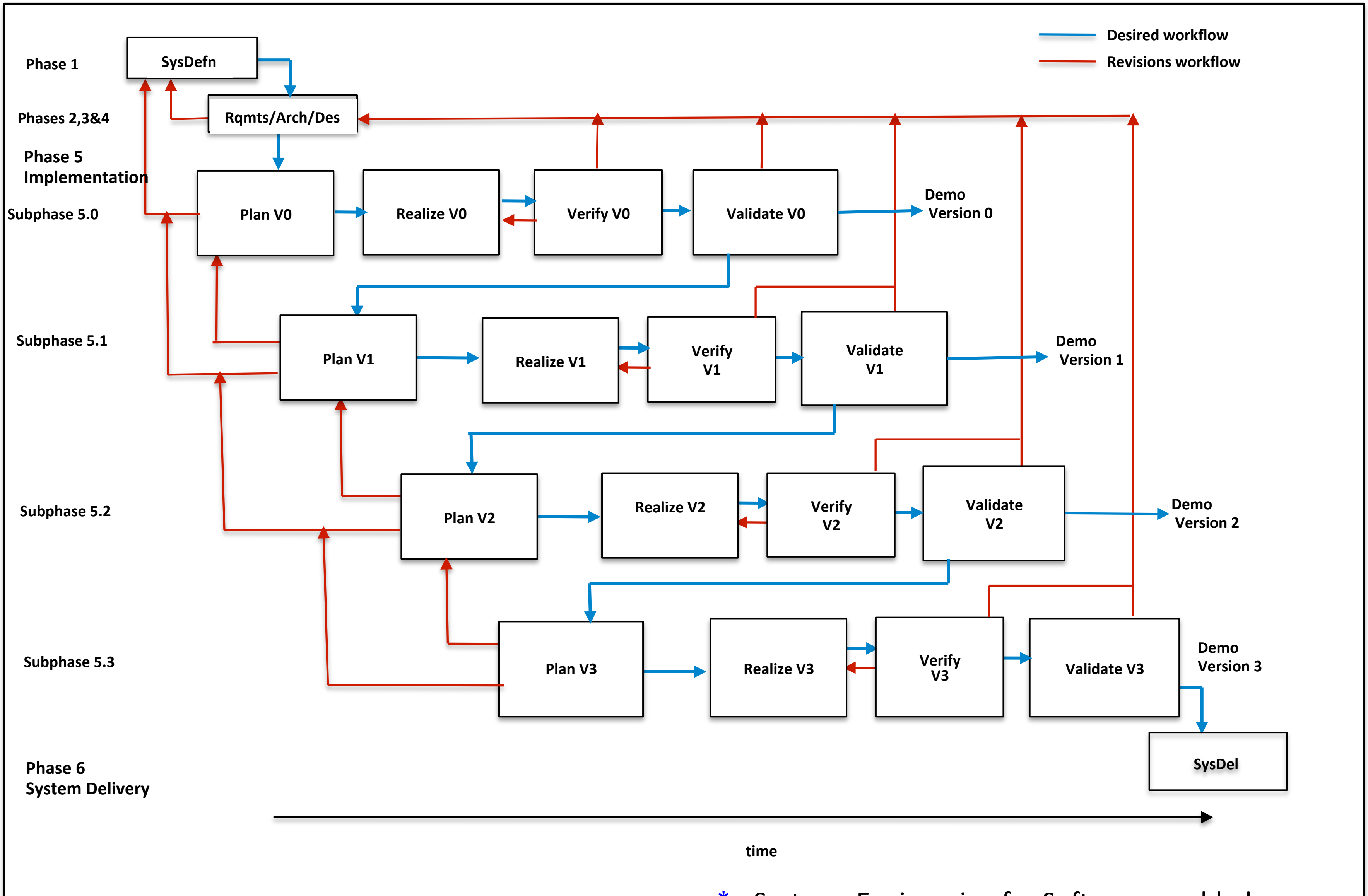
- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

The I^3 MBSE – M&SBSE development model

- I^3 is the integrated-iterative-incremental model for system development
- An MBSE approach is used to model requirements, architecture, and design
- An M&SBSE approach is used for system implementation
  - With coordinated concurrent iterative-incremental implementation of hardware and software elements
  - And development of software-enabled physical elements

NOTE: I^3 is a work in progress

Figure 5.2 The I^3 system development model
Source: Figure 5.2*

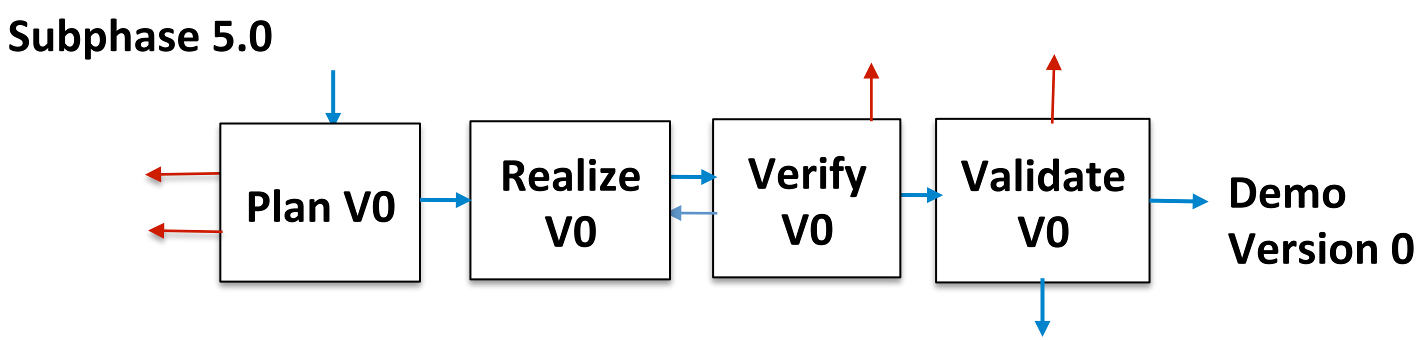


* Systems Engineering for Software-enabled Physical Systems; R. Fairley, Wiley 2019

I^3 ATM implementation subphases

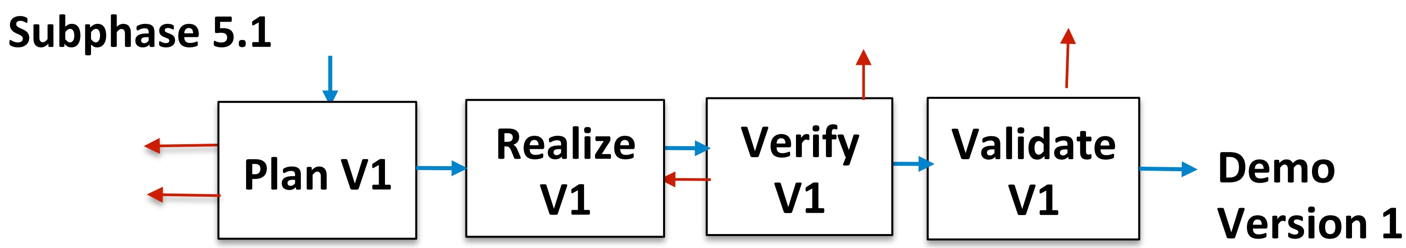
- Subphase 5.0: A black-box simulation model of ATM elements and interfaces
- Subphase 5.1: Secure Customer Authentication capability
- Subphase 5.2: Secure Session Termination capability
- Subphase 5.3: Balance Query capability
- Subphase 5.4: Cash Withdrawal capability
- Subphase 5.5: Funds Deposit capability
- Subphase 5.6: Funds Transfer capability

I^3 Phase 5 subphase 5.0



Version 0 (Subphase 5.0) is a black box simulation model of a system or subsystem
for example, elements and interfaces for the ATM card reader, display screen, keypad, cash safe, cash dispenser, funds depository, and printer

I^3 Phase 5 subphase 5.1



When completed Subphase 5.1 provides Secure Customer Authentication using a real card reader, keypad, display screen, and user interface with black box simulations of the other system elements and interfaces (Version 1)

- The system is complete when the last simulated system element and/or simulated interface is replaced with a real element and interface
- Early deliver of a real/simulated system may be provided for evaluation, training, and perhaps limited use

Incremental V&V: Formal vs. Informal

- Formal V&V is accomplished by V&V specialists
  - Typically in a separate department
  - Or in a separate organization for IV&V
  - And is applied to a deliverable system, a subsystem, or a system element
- Informal V&V (iv&v) is accomplished by SEs, disciplinary engineers, specialty engineers, and domain experts
  - Informal does not mean haphazard
  - Techniques include traceability analysis, simulation scenarios, prototyping, inspections, demonstrations and reviews

Some V&V specialists are not prepared to accomplish formal incremental V&V

Some aspects of I^3

- Hardware and software development can proceed concurrently
  - With coordinated development of the interfaces
- Each implementation subphase implements a system increment/ capability
  - by replacing simulated system elements with realized system elements (hardware, software, interfaces)
- Different disciplinary and specialty engineers can use their preferred development methods
  - for example, vee increments and iterative capabilities

> - Real system elements and interfaces replace simulations when they become available
> - Demonstrations of progress occur on a monthly basis and plans are made for the coming month

Some I^3 details

- An individual or team maintains the integrity and fidelity of the system simulation
- An individual or team coordinates development of hardware/ software interfaces with the system simulators and development engineers

  and is the keeper of the shared Interface Control Document,

    which is under configuration control
- Disciplinary engineers use the evolving simulated/realized system as a test stand for their work and replace simulated elements with realized elements when they become available

Some MBSE – M&SBSE Issues

- Restructuring organizational policies and procedures
- The relationship between development and formal V&V
- Redoing system development and documentation procedures
- Maturity and compatibility of the development infrastructure, development environment, and development tools
- Discrepancies between simulated and real system elements and interfaces (fidelity and integrity)
- Retraining of disciplinary and specialty engineers (e.g., hardware engineers of various kinds and software engineers)
- Scaling of simulation models
- Simulation specialists needed
- Developing and maintaining the system simulation

“Overhead” issues

- “Overhead” issues:
  - Simulation specialists are needed
  - The initial system simulation must be developed
  - Integrity and fidelity of the simulation must be maintained as the simulated/real system evolves

“Overhead” issues (2)

- Experience to date indicates that the overhead of the M&SBSE approach, when combined with MBSE:
  - Can improve communication among engineers
  - Eliminates unneeded documentation
  - Reduces rework by identifying and resolving issues
    - ✓ rework is the bane of system development
  - The result is more efficient and more effective system development, which results in higher quality systems and happier stakeholders and system developers
    - ✓ more is saved than is invested

The evidence is anecdotal at this point in time

Some MBSE – M&SBSE Cautions

- Don't use new methods and tools for present development processes
  - Rethink analysis, architecture, design, and implementation and the relationships among them
  - Rethink documentation requirements and needed documents
- Invest in training and pilot projects
- Don't go "big bang" with MBSE or M&SBSE
  - New techniques and technologies must be phased in incrementally

Many new initiatives have failed for lack of an effective technology transfer strategy and plans

US DOD Technology Readiness Levels
https://techlinkcenter.org/technology-readiness-level-dod/

Technology Readiness Level 1: Basic principles observed and reported

Technology Readiness Level 2: Technology concept and/or application formulated

Technology Readiness Level 3: Analytical and experimental critical function and/or characteristic proof of concept

Technology Readiness Level 4: Component and/or breadboard validation in a laboratory environment

Technology Readiness Level 5: Component and/or breadboard validation in a relevant environment

Technology Readiness Level 6: System/subsystem model or prototype demonstration in a relevant environment

Technology Readiness Level 7: System prototype demonstration in an operational environment

Technology Readiness Level 8: Actual system completed and qualified through test and demonstration

Technology Readiness Level 9: Actual system proven through successful mission operations

Agenda for System Implementation

- A brief review of training webinars 1, 2, 3, and 4
- The Implementation process
- The Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

Roles of systems engineers in system implementation

- Planning and coordinating system implementation*
- Coordinating development of hardware and software
- Ensuring sufficiency of verification and validation
- Providing liaison of disciplinary and specialty engineers with system architects and system stakeholders
- Facilitating introduction of MBSE and M&SBSE methods and tools

*See 15288 Clause 6.3 for Technical Management processes

Note

- Systems engineers facilitate system implementation
  - Implementation and integration of system elements is accomplished by disciplinary engineers and specialty engineers
- Systems engineers may also apply their engineering disciplines to participate as disciplinary or specialty engineers
  - But they are playing different roles if they do so

A common complaint

A common complaint:

the systems engineers disappear during system implementation, final V&V, transition, and delivery

Some questions:

1. Are SEs needed after initial analysis and design?

   what is the cost/benefit ratio?
2. If SEs are not present, who will facilitate solving the larger problems and provide liaison with architects and stakeholders?
3. Is scarce expertise needed elsewhere?
4. Is project management sufficient in later phases?

Agenda for System Implementation

- Brief review of training webinars 1, 2, 3, and 4
- Systems implementation processes
- Integration process
- Some system implementation methods
- MBSE and M&SBSE
- The I^3 MBSE – M&SBSE system development model
- Roles of systems engineers in system implementation
- Comments and questions

Questions? Comments?

- Contact information:

  dickfairley@gmail.com

  john.clark@incose.org

  gabriela.coe@incose.org