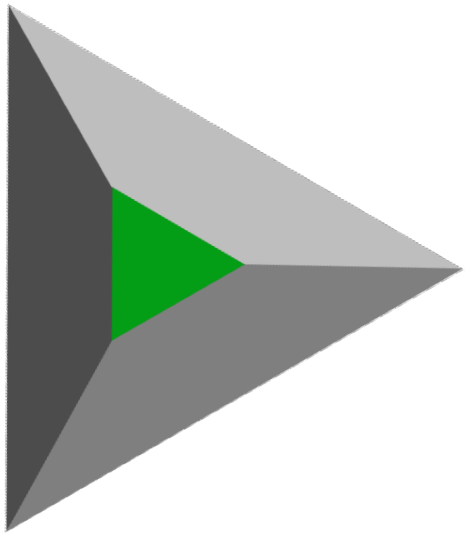


Velentium

Velentium = **Vel**ocity + Mom**ent**um + Ingen**ium**

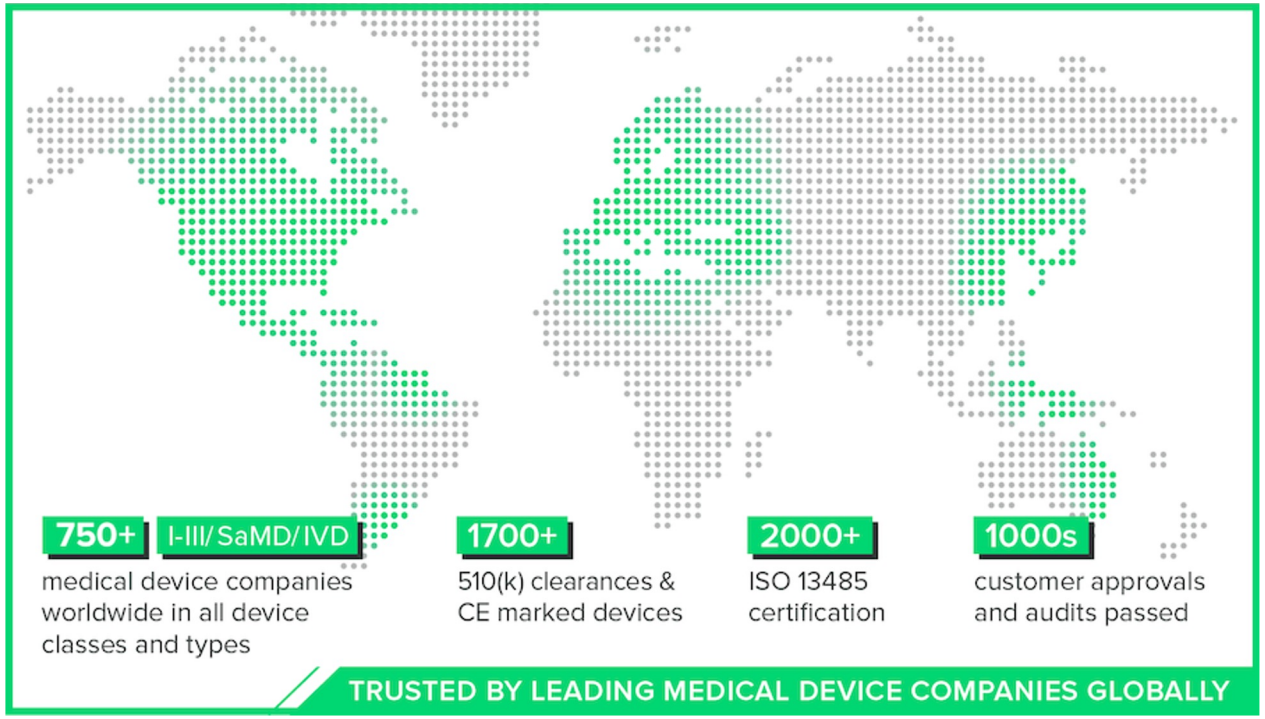
MEDICAL DEVICE QUALITY IS ALL WE DO, AND WE'RE ALWAYS AHEAD OF THE GAME.

1.5M years industry experience

522k podcast listeners

182k+ look to us for the latest in quality

#1 blog and podcast in the industry



"Best eQMS I have ever used..."

G2 CROWD LEADER MEDICAL QMS SOFTWARE ★★★★½ Since Spring 2021

G2 CROWD LEADER QMS SOFTWARE ★★★★½ Since Winter 2019

This is the easiest eQMS I have used in the 20 years I have been in the Medical Device Industry. ***It is simple, intuitive and easy to use...*** We are successfully implementing a Quality Culture.

- Director of Regulatory Affairs & Quality Assurance

"Modern QMS Software and Outstanding Customer Service."

★★★★★

"Demystifying QMS and Regulatory Requirements"

★★★★★

"Makes your QMS Simple and Effective"

★★★★★

greenlight guru

ce Engineering: Your IP. Designed and Built.

Extending IEC 62304 principles

Applying development process principles to other* project disciplines and functions

*Other than SOFTWARE. Note, all SMALL CAPS are defined in IEC 62304, Part 3.

Agenda – Extending IEC 62304 principles

Medical Device – a quick background

IEC 62304 – a background

- What is it?
- Why use it to extend?

Extending IEC 62304

- A look at common aspects
- Adopting basic principles (when, why, how*)

Focus on Development Process

- Requirements through Release

Stakeholders

- Program Directors, Project Managers, Systems and Process, Technical Leads and Engineers

*Low-level process and sub-process implementation details and tool choices are up to the individual MFGs. This ensures flexibility and scalability of these processes.

Medical Device – a Background

- Medical Devices are categorized in Classes (I, II, III)
  - Covers a huge range of diagnostic and delivery devices
  - Software permeates a significant portion of devices (active devices)
- Reliability and Safety (Risk associated) of Software of vital concern
  - Align with regulatory requirements across device classes.
- Response to this vital concern – IEC 62304
  - Processes to demonstrate compliance to requirements

What is IEC 62304?

Titled: “Medical Device Software – software lifecycle processes”

- Original Release in 2006 with an Amendment in 2015.
  IEC 62304:2006+AMD1:2015
- Functional Safety Standard
- Covers safe design and maintenance processes of software
- Provides PROCESSES, ACTIVITIES, and TASKS to ensure a safe MEDICAL DEVICE.

Applies to Development and Maintenance when

- SOFTWARE is itself a medical device
- SOFTWARE is an embedded or integral part of medical device

*IEC: International Electrotechnical Commission

IEC 62304 – Standards relationship

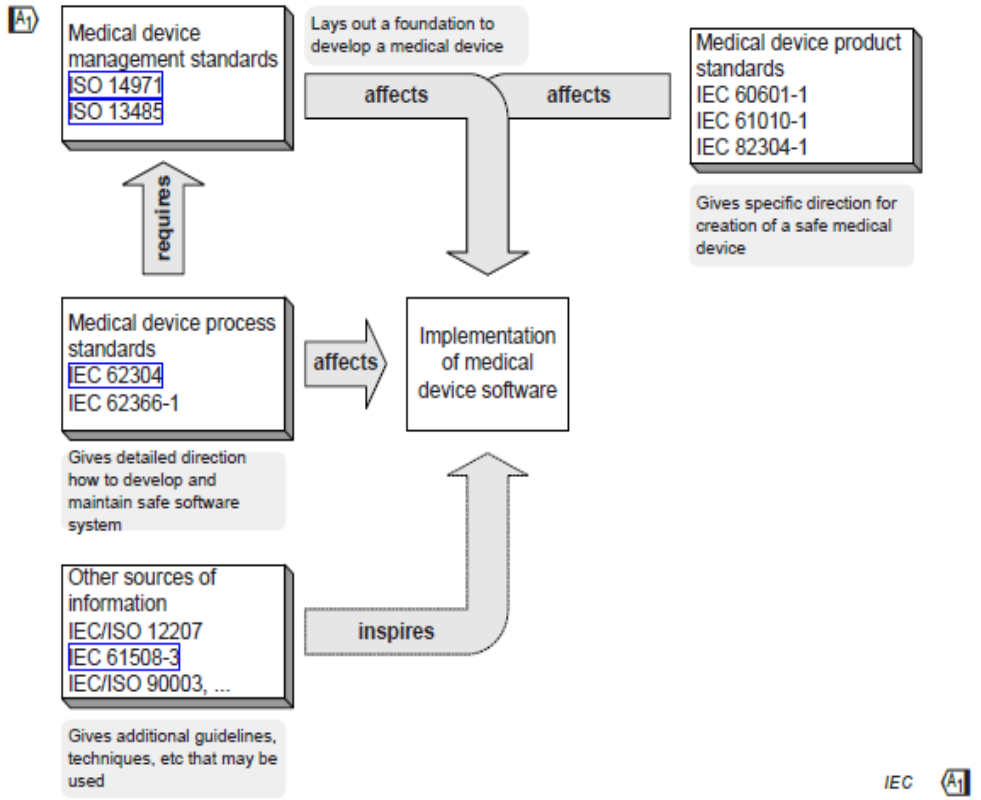
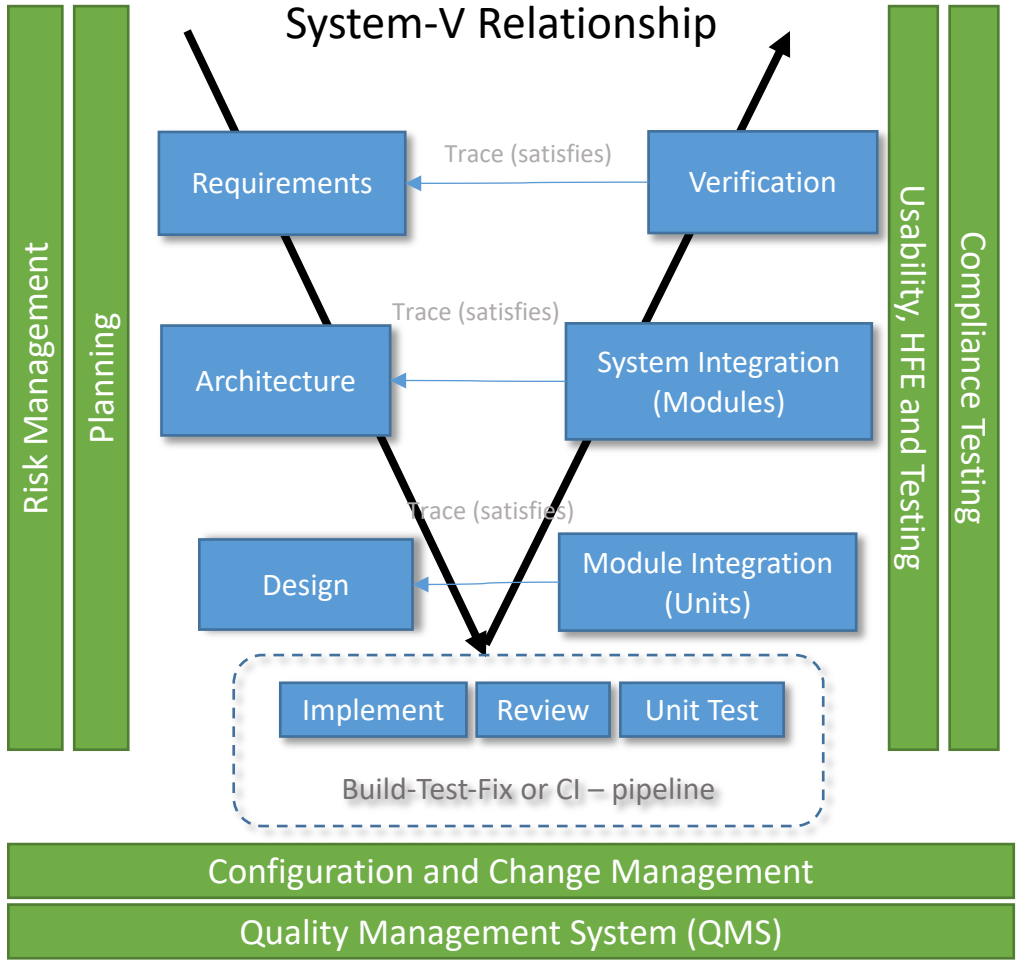


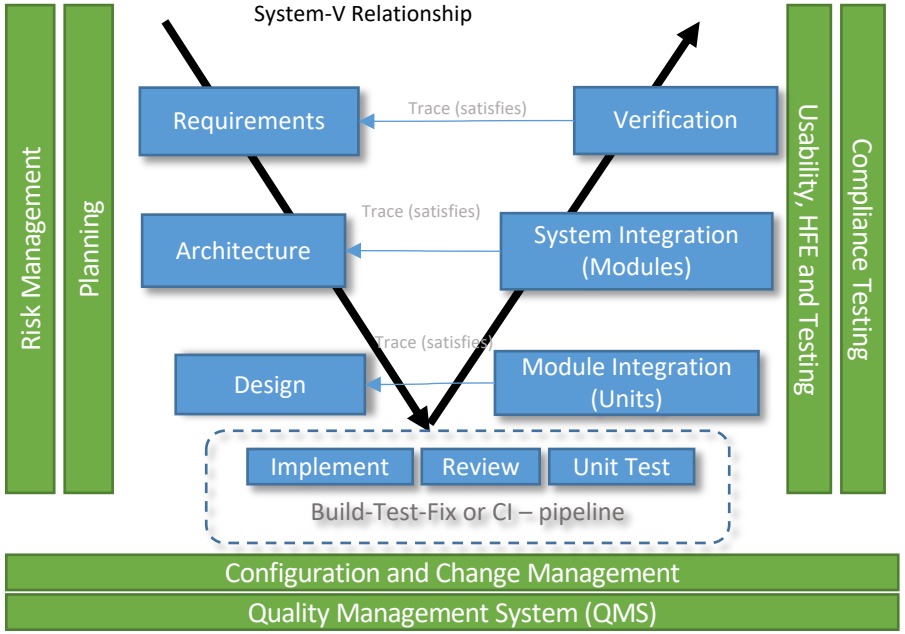
Figure C.1 – Relationship of key MEDICAL DEVICE standards to IEC 62304



IEC 62304 - Organization

Core Processes

- **Development Process**
- Maintenance Process
- Risk Management Process
- Configuration Management Process
- Problem Resolution Process



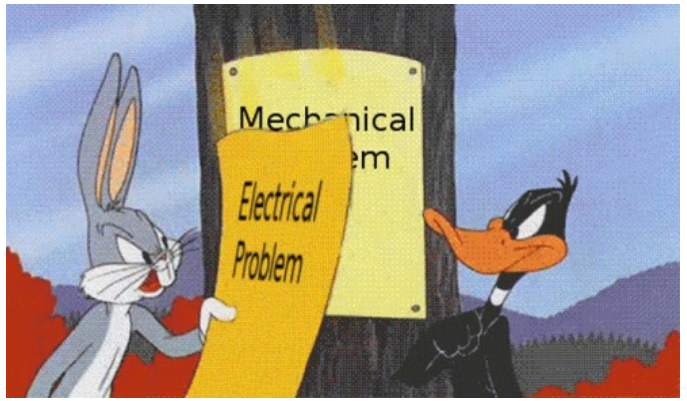
Nine Parts -

- **Part 1:** Scope.
- **Part 2:** Normative references.
- **Part 3:** Terms and definitions.
- **Part 4:** General requirements.
- **Part 5: Software development process.**
- **Part 6:** Software maintenance process.
- **Part 7:** Software risk management process.
- **Part 8:** Software configuration management process.
- **Part 9:** Software problem resolution process.

Why IEC 62304?

Why extend the IEC 62304 principles to Electrical/Mechanical?

- There is no singular guidance or standard that describes or prescribes lifecycle processes for other disciplines or functions like Electrical and/or Mechanical.
- Most development processes are derived from a mixture of other guidances and best practices
  - Leads to misaligned terms and confusing PROCESS definitions.
  - Leads to “siloed” development practices.
  - Leads to incomplete planning of ACTIVITIES and improper TASK dependencies



Why IEC 62304?, cont.

Extending IEC 62304 principles should

- Provide consistency in PROCESS and term definitions
- Provide appropriate planning of ACTIVITIES and TASKS
- Aide in a collaborative development process – break down “siloes”
- Provide a disciplined, focused, requirements and risk based approach to development of medical devices

What this does NOT mean (not mandatory!)

- Extending these principles **does not** mean compliance to IEC 62304 as it’s not required.
- This is a guidance only and voluntary for Electrical/Mechanical

Extending IEC 62304

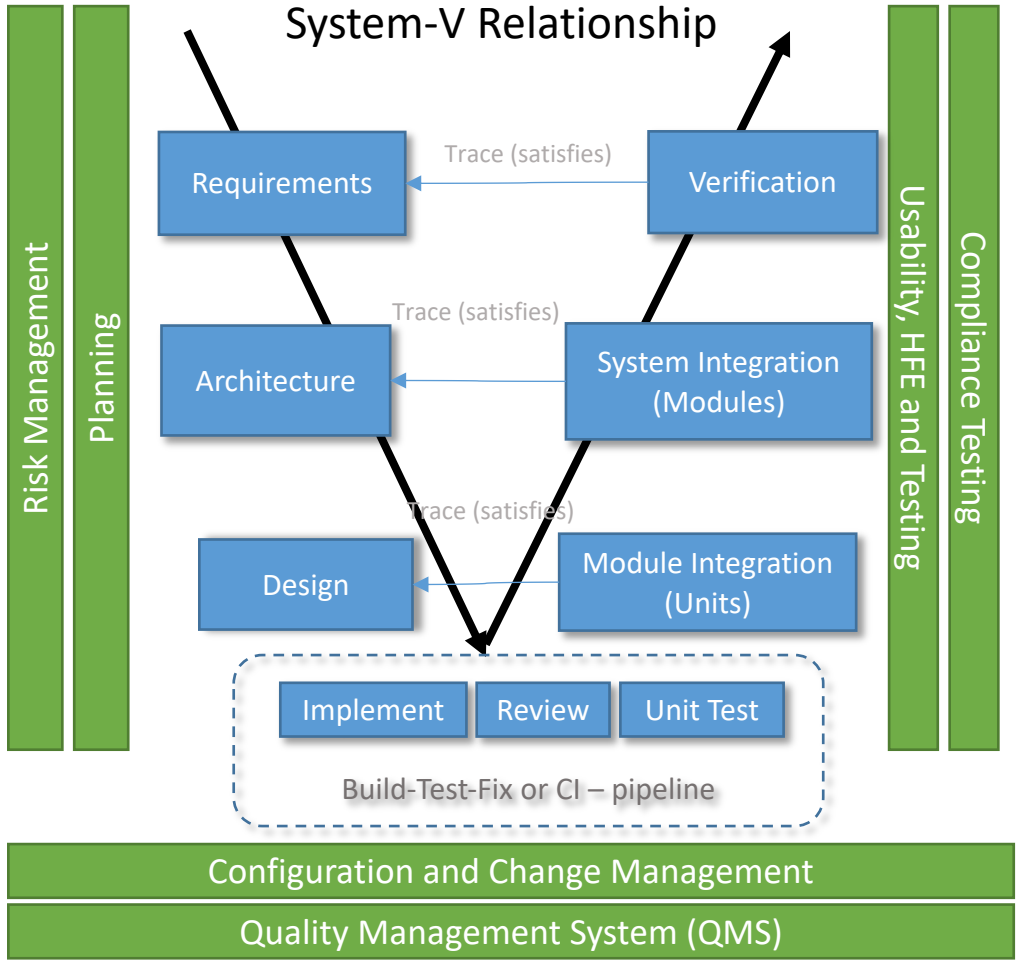
- Review: Common Elements
- In-depth: Software Safety Classification
- In-depth: Adoption of Development Process and Extending to Electrical/Mechanical

IEC 62304 – Common Elements

Quality Management System Compliance

- International standard for QMS in Medical Device is ISO 13485.
- Other national regulations are allowed – US, FDA 21 CFR 820
- Basis for all Medical Device MFGs
- Covered in Part 4.

All common elements shall be captured through Planning



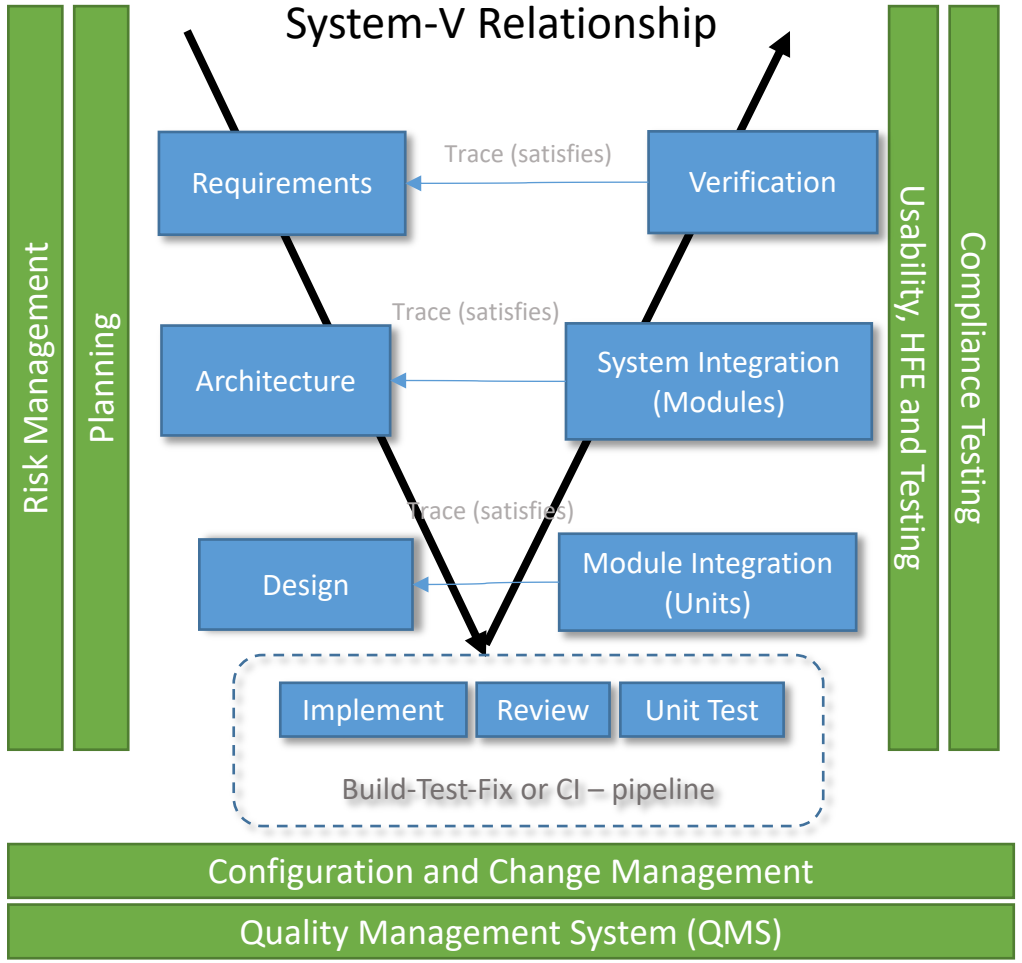
IEC 62304 – Common Elements

Risk Management Compliance

- Compliance with ISO 14971:2019 is required
- Included Hazard analysis and Failure Mode Analysis
- Covered in Part 4 and Part 7

Configuration and Change Management

- Documentation and Implementation of configuration items
- Traceability and Verification
- Cover in Part 6, 8 and 9



IEC 62304 – Software Safety Classification

- Determination of this impacts the Software Development Process
- There are 3 safety classes for software
  - Class A: No injury or damage to health is possible.
  - Class B: Injury is possible, but not serious.
  - Class C: Death or serious injury is possible.
- Part 4.3 provides the flowchart on how to assign the safety class

The SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which results in unacceptable RISK **after consideration** of RISK CONTROL measures **external to the SOFTWARE SYSTEM** and the resulting possible HARM is death or SERIOUS INJURY.

NOTE 1 External RISK CONTROL measures can be **hardware**, an independent SOFTWARE SYSTEM, health care procedures, or other means to minimize that software can contribute to a HAZARDOUS SITUATION.

A1 a) The MANUFACTURER shall assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the RISK of HARM to the patient, operator, or other people resulting from a HAZARDOUS SITUATION to which the SOFTWARE SYSTEM can contribute in a worst-case-scenario as indicated in Figure 3.

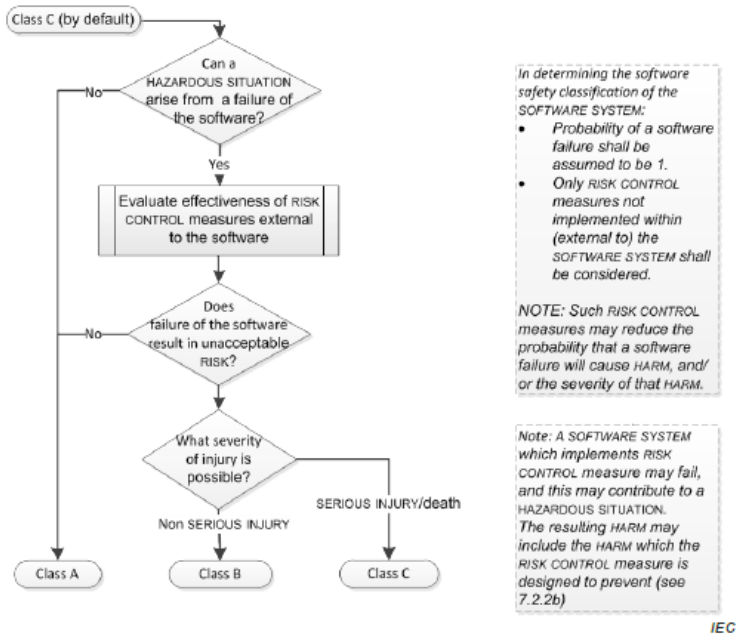
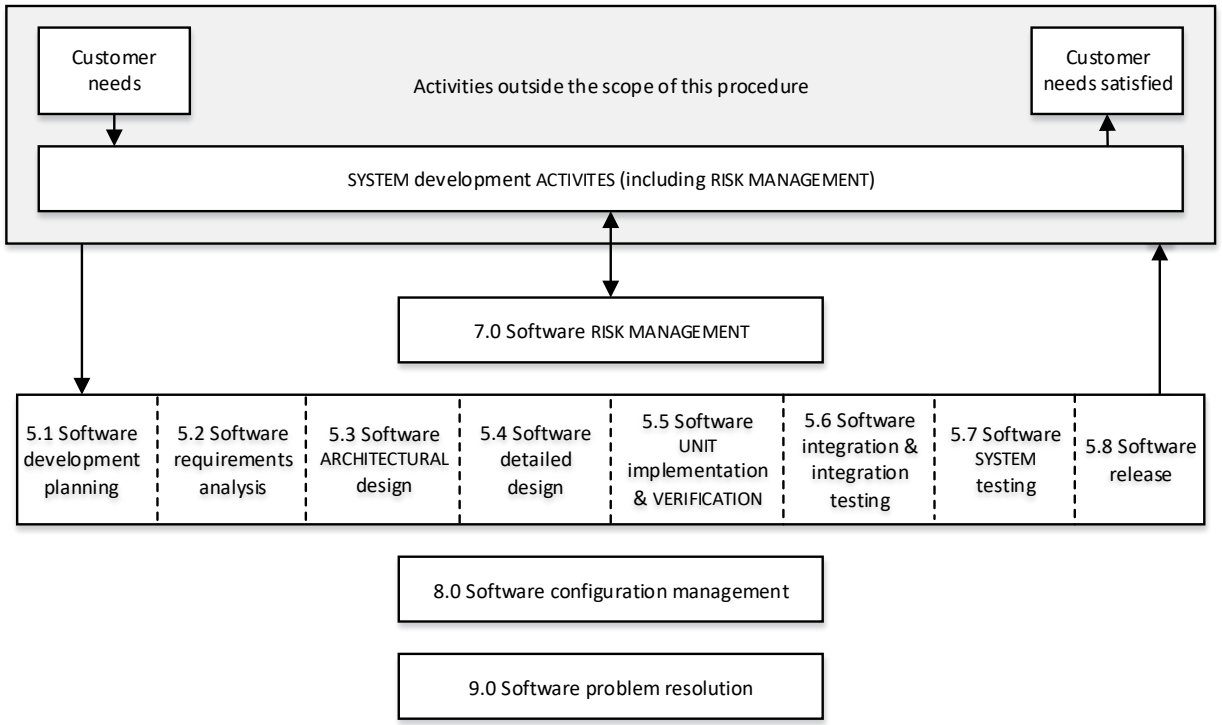


Figure 3 – Assigning software safety classification

IEC 62304 – Development Process



Software documentation	Class A	Class B	Class C
Software development planning	X	X	X
Software requirements analysis	X	X	X
Software architectural design		X	X
Software detailed design			X
Software unit implementation	X	X	X
Software unit verification		X	X
Software integration and integration testing		X	X
Software system testing	X	X	X
Software release	X	X	X
X - required			

Note: identify and differentiate between activities and tasks performed vs evidence of compliance as DHF artifacts It is always a best practice to document and review all activities and tasks appropriately but they should always be performed.

IEC 62304 – Key Terms and Definitions

Extending the following IEC 62304 terms

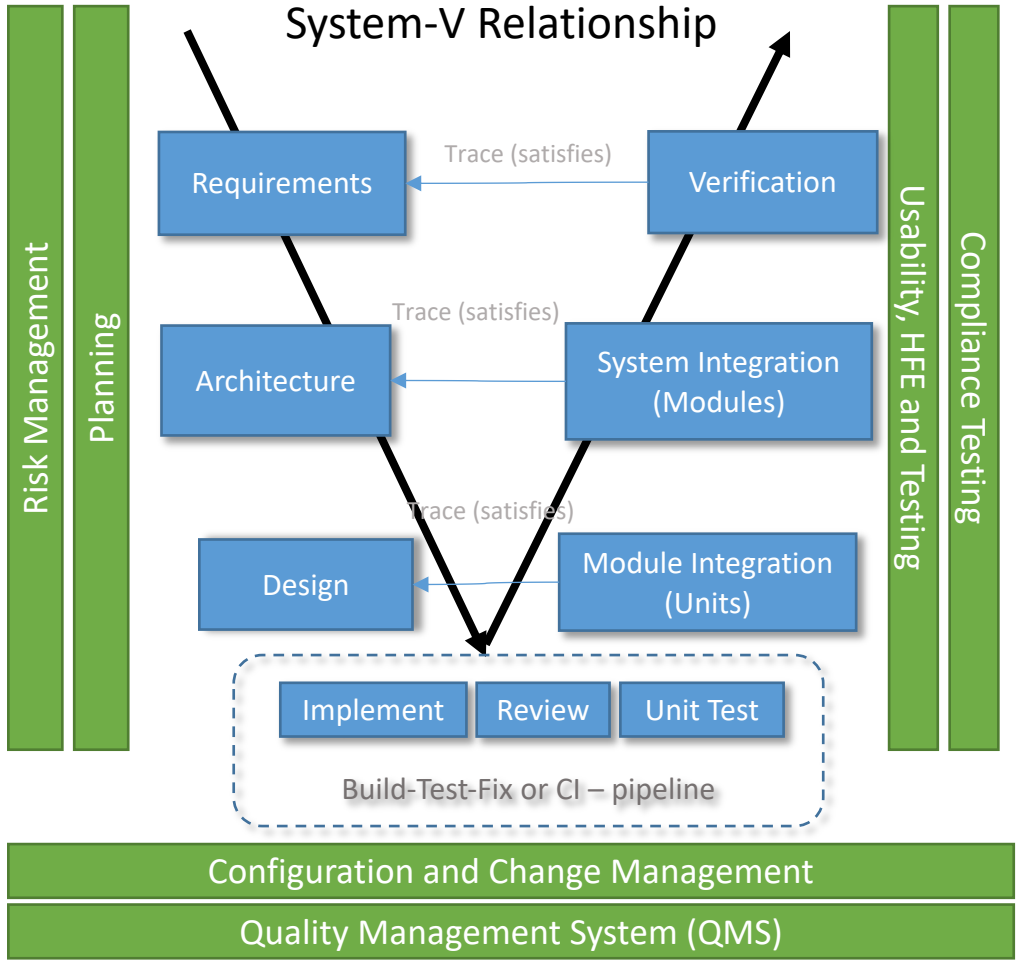
- SOFTWARE UNIT
- SOFTWARE ITEM
- SOFTWARE SYSTEM

IEC 62304 terms	Software	Electrical/Mechanical
UNIT	A single function or method in code	A component (active or passive) on a schematic or assembly that interfaces to or aides an action or behavior
ITEM	A collection of UNITS that identify a behavior	A Collection of components that identify a behavior
SYSTEM	A collection of ITEMS to achieve an objective that meets user needs	

IEC 62304 –Adopting and Extending Development Process

Part 5 - Development Process

- 5.1: Development planning.
- 5.2: Requirements analysis.
- 5.3: Architectural design.
- 5.4: Detailed design.
- 5.5: Unit implementation and verification.
- 5.6: Integration and integration testing.
- 5.7: System testing.
- 5.8: Release.



IEC 62304 –Adopting and Extending Development Process

5.1 – Planning –

SOFTWARE DEVELOPMENT LIFE CYCLE MODEL – Conceptual structure spanning the life of the **software** from definition of its requirements to its release (3.24)

As part of planning, the Technical Lead should consider and collaborate with Project Management to:

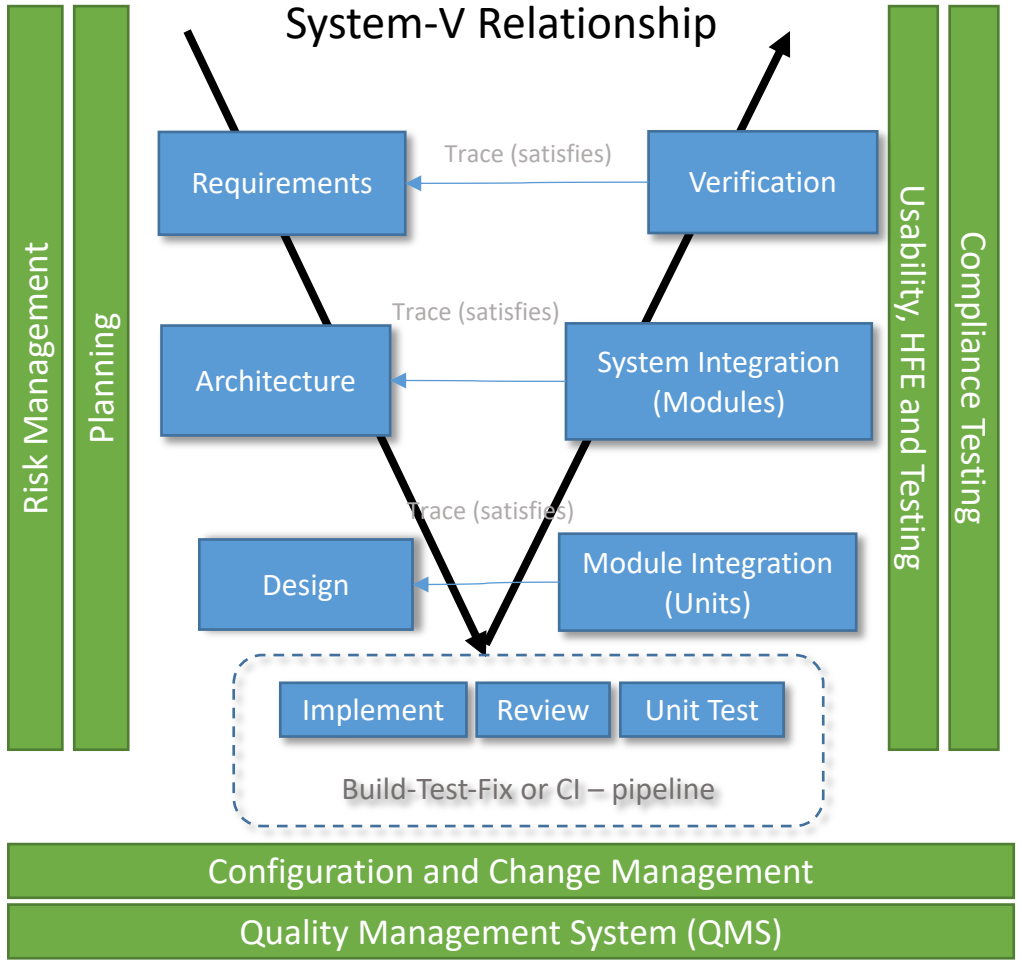
Identify the PROCESS, ACTIVITIES and TASKS involved in development of a medical device

Describe the sequence of and dependency between ACTIVITIES and TASKS

Identify the milestones at which of the completeness of specified deliverables is verified

Important: for Electrical/Mechanical timescales vary and can conflict with established cadences. Ex: Agile Model utilizes a 2-week sprint mechanism

- Describe the "How" and "When"
  - Includes ACTIVITIES and TASKS involved in the development PROCESS
  - Identify development environments and tools
  - Identify configuration and change management
- Extended planning to Testing
  - Create a plan for Testing
  - Identifying various methods – model-based, design analysis, evaluation and measurement, inspection, etc.
  - Identify test environments, test equipment and sample size(s)

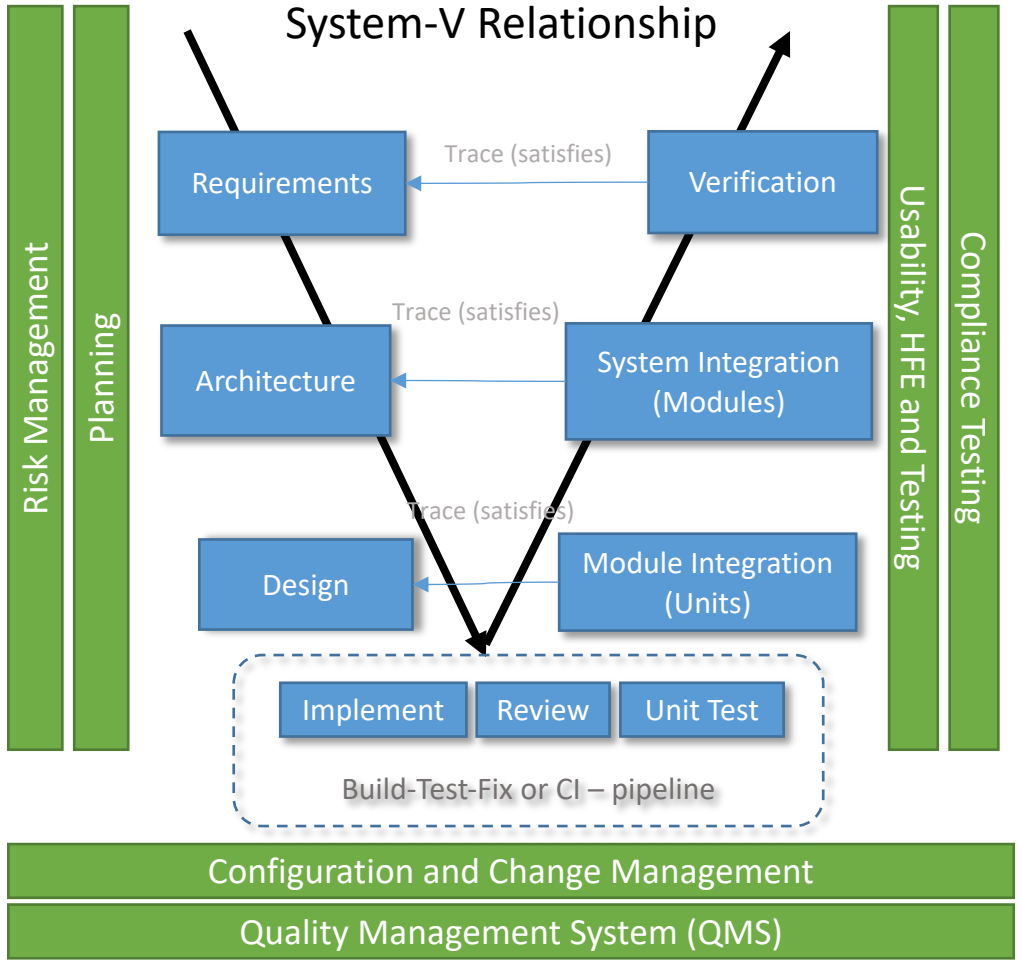


IEC 62304 –Adopting and Extending Development Process

5.2 – Requirements Analysis

- Review and Analysis System Requirements
  - Including mitigations from Hazard analysis (RISK CONTROL) as Requirements
- Identify and Refine further into a Technical Specification
  - Input specification - can be called a SRS (software requirement specification) or generically sub-system requirements.
  - Results in a prescriptive list of responsibilities the SYSTEM owns and is refined by the ARCHITECURE. Provides traceability to SYSTEM Testing per Plan (5.1)

Collaboration and compromise between Systems, other Technical Leads and Engineers. Systems is the stakeholder



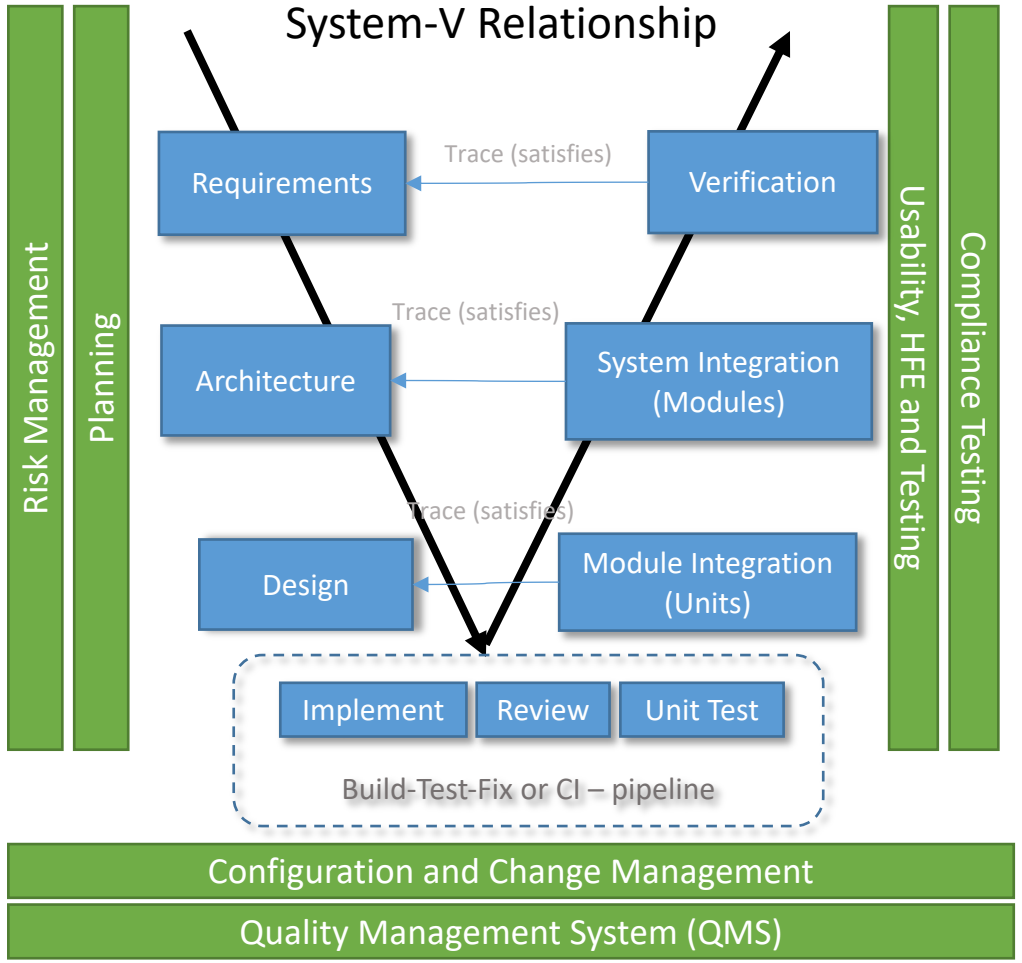
IEC 62304 –Adopting and Extending Development Process

5.3 ARCHICTURAL detail design

- ARCHICTURAL design
  - Descriptive decomposition based on Requirement Analysis
  - Identify and segregate the SYSTEM into ITEMS
    - Further refining sub-systems into functional blocks
    - Capabilities with external and internal interfaces
  - Identifies behavior and responsibilities
    - Describes mechanism, models, concepts
    - Defined at the block diagram level.
  - Identify Technical Processes
    - PCBA fab (flex circuits), custom packing of ICs
    - Injection molding vs 3d printing, welding

Compromise between Systems, Technical Leads and Engineers. System is the stakeholder

Note: This is not detailed drawings or schematics for Electrical/Mechanical.

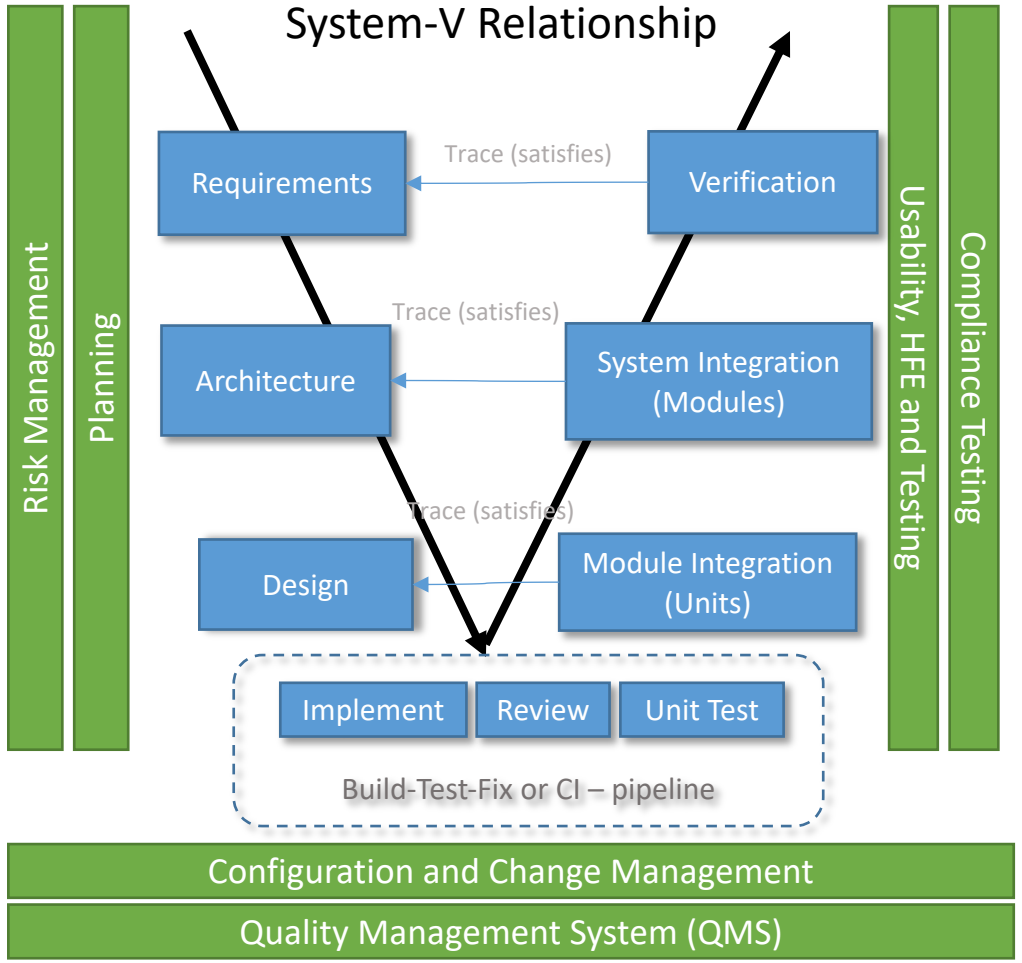


IEC 62304 –Adopting and Extending Development Process

5.4– detail design

- Detailed Design
  - Descriptive decomposition based on Requirement Analysis
  - Identify and subdivide ITEMS into UNITS
  - Further refine external and internal interface segregation.
  - Include transient and persistent data details
  - Include detailed flows and sequences
  - Include initial failure mode analysis

- For Electrical/Mechanical, this we where detail drawings and/or schematics and layout are created.
- Also includes design analysis, review of models, change history, Gerber, BOM, etc.
- An initial Output specification (think datasheet: min, nom, max) should be generated. Current, voltage, power, force Collaboration between Engineers. Stakeholders are Technical Leads.



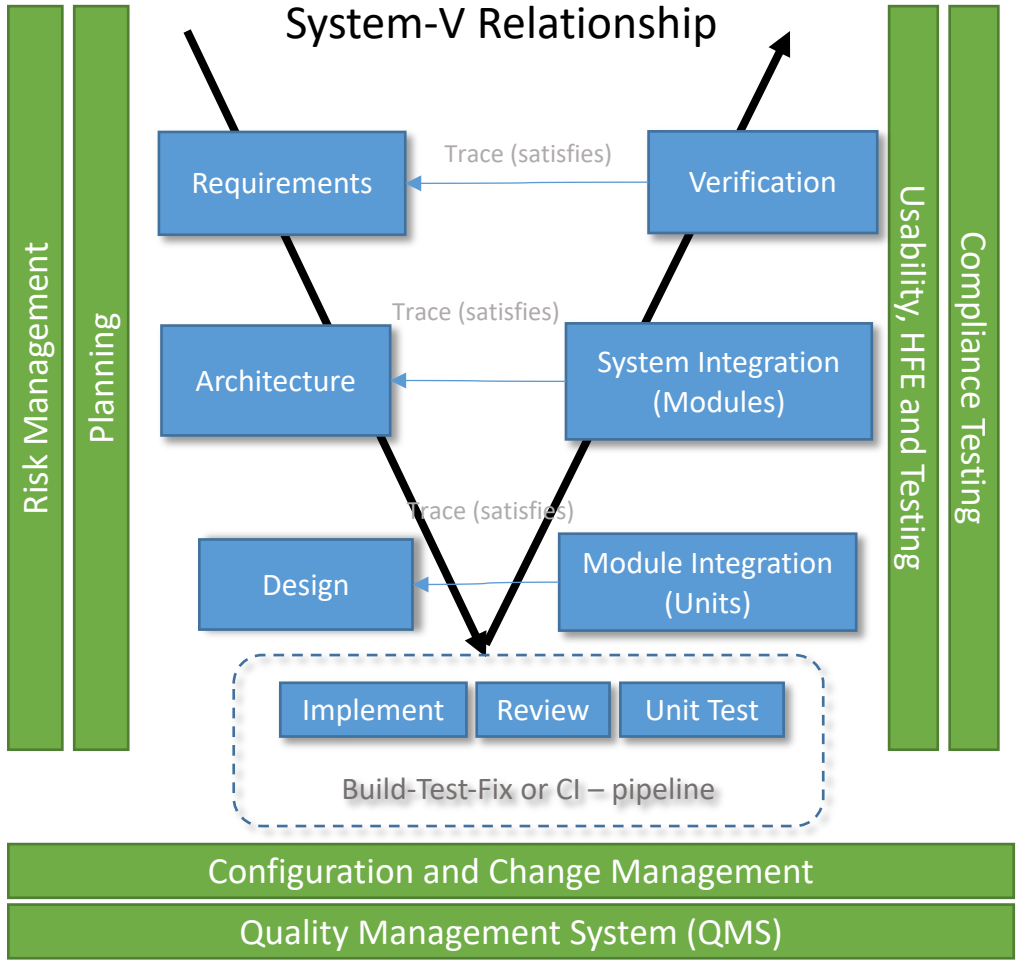
IEC 62304 –Adopting and Extending Development Process

5.5 – Implementation

Building phase

- For Electrical/Mechanical this is where a design (paper) is translated into a physical object.
- PCBA, connectors, buttons, housing, cables, etc.
- Level of implementation should be associated with the plan (5.1). Breadboard, prototype, 3d-printing.
- Refine the technical processes – PCBA fab (flex circuits), injection molding, welding, etc.

Collaboration with Engineers and Tech Leads.
Important to refine dependencies per Plan (5.1)



IEC 62304 –Adopting and Extending Development Process

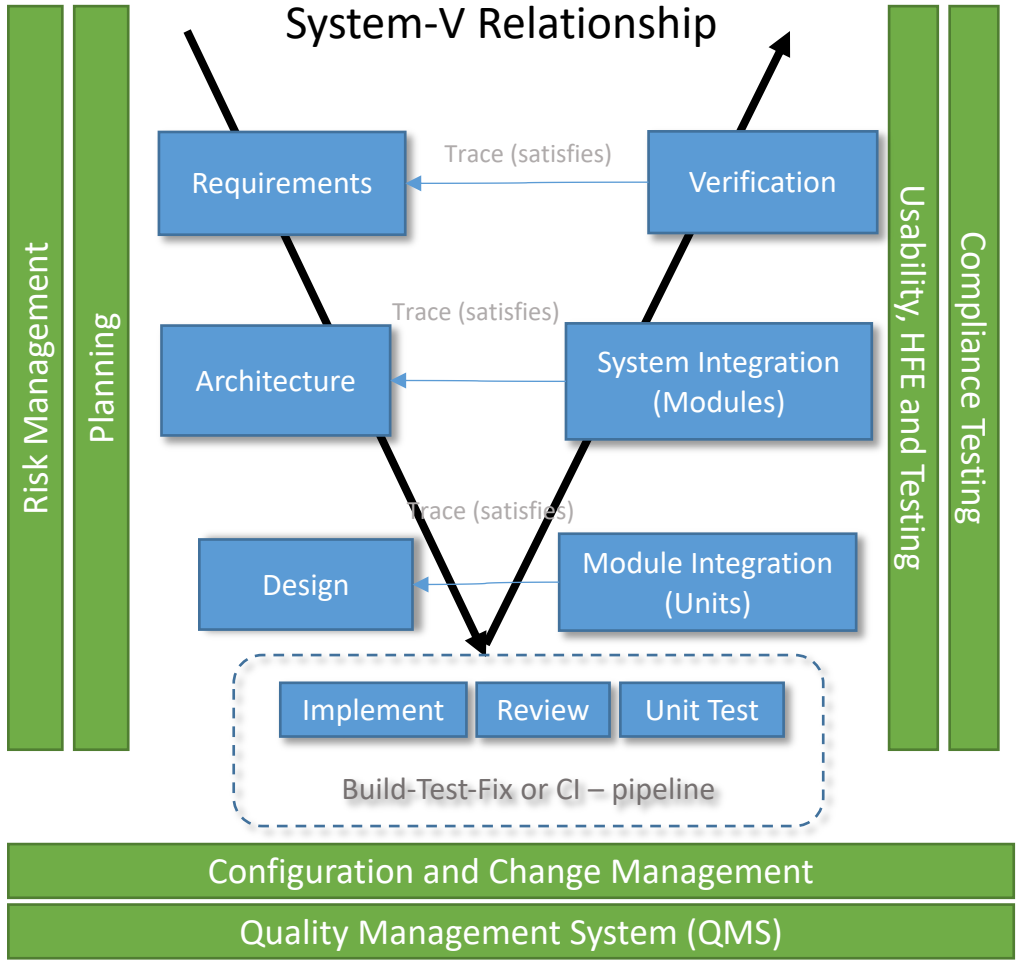
5.5 – Implementation

Testing phase

Establish strategies, methods and procedures for verifying UNITS

- Testing at the UNIT level – includes board bring-up ACTIVITIES, tolerances, stack-up analysis or measurements.
- Confirmation or clarification of Output specification
- Identify and review verification protocol

Collaboration with Engineers and Tech Leads. Important to refine dependencies per Plan (5.1)



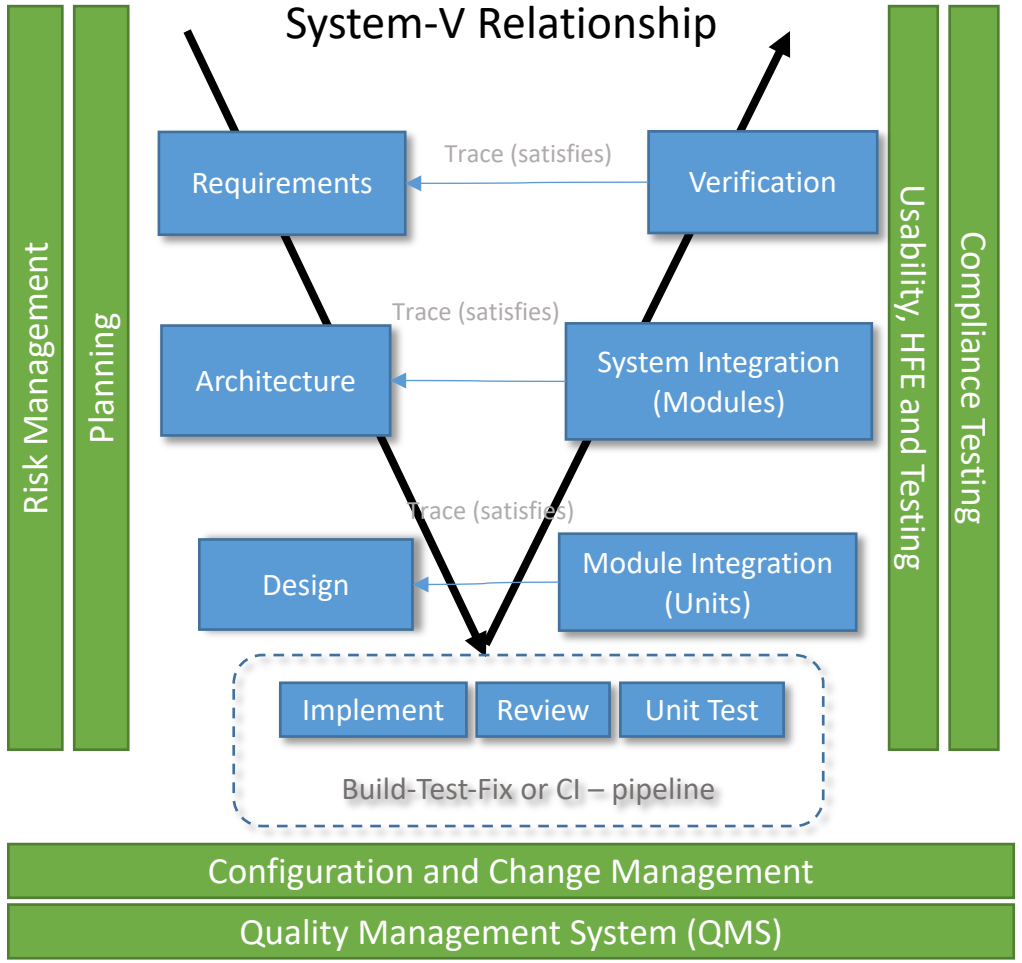
IEC 62304 –Adopting and Extending Development Process

5.6: Integration

Integration

- Establish strategies, methods and procedures for integrating of UNITS into ITEMS
- Integration of ITEMS for Electrical/Mechanical is established if there is a single PCBA design and implementation.
- Integration between functions, Electrical, Mechanical and Software shall be part of SYSTEM testing
- Document test configuration, acceptance criteria as part of Test procedures per Test plan (5.1)

Collaboration between the Engineers. Stakeholders are Systems and Technical Leads.

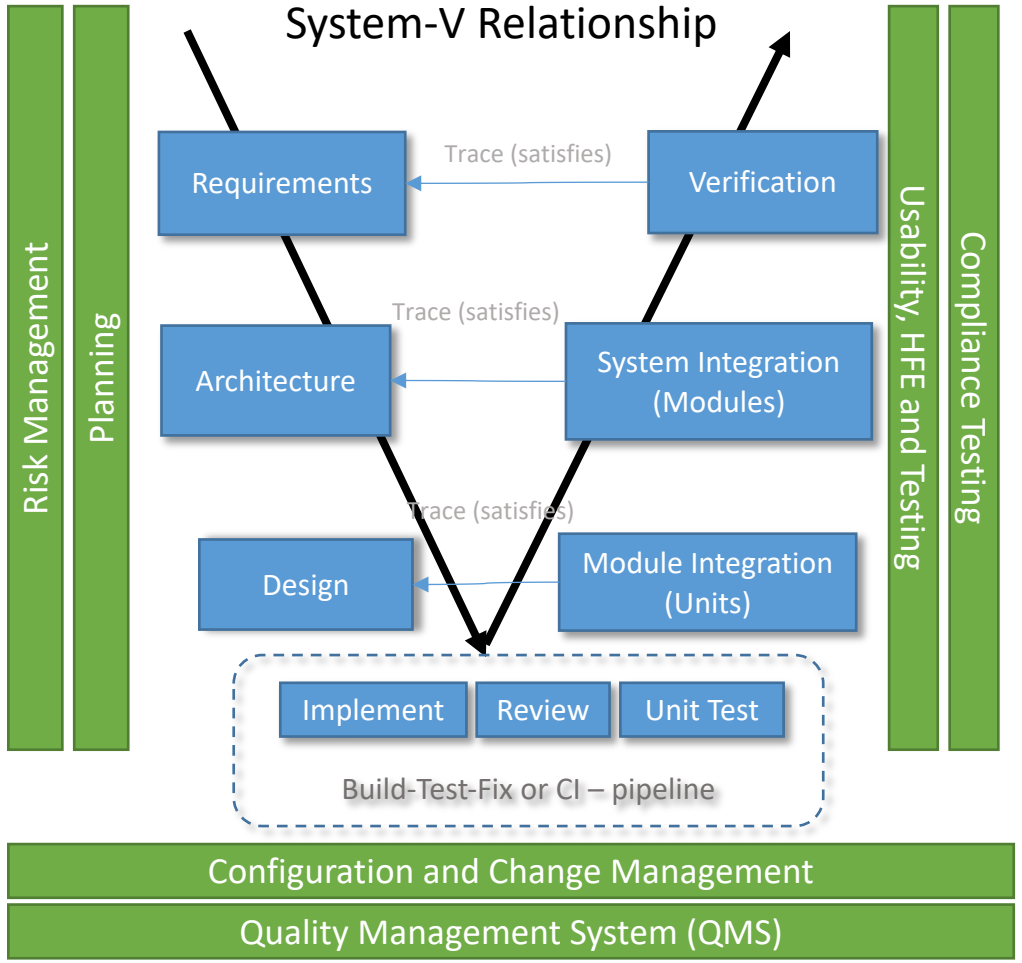


IEC 62304 –Adopting and Extending Development Process

5.7: System Testing

- Refine strategies, methods and procedure for the SYSTEM testing per Test plan (5.1)
- Document test configuration, acceptance criteria as part of Test procedures per Test plan (5.1)
- Execution of Test procedures per Test plan (5.1) with traceability from ITEMS to SYSTEM behaviors.
- Provide TRACEABILITY to the sub-system REQUIREMENTS
- Can be considered as dry-run for formal PRODUCT VERIFICATION

Collaboration between the Testers and Engineers.
Stakeholders are Systems and Technical Leads.



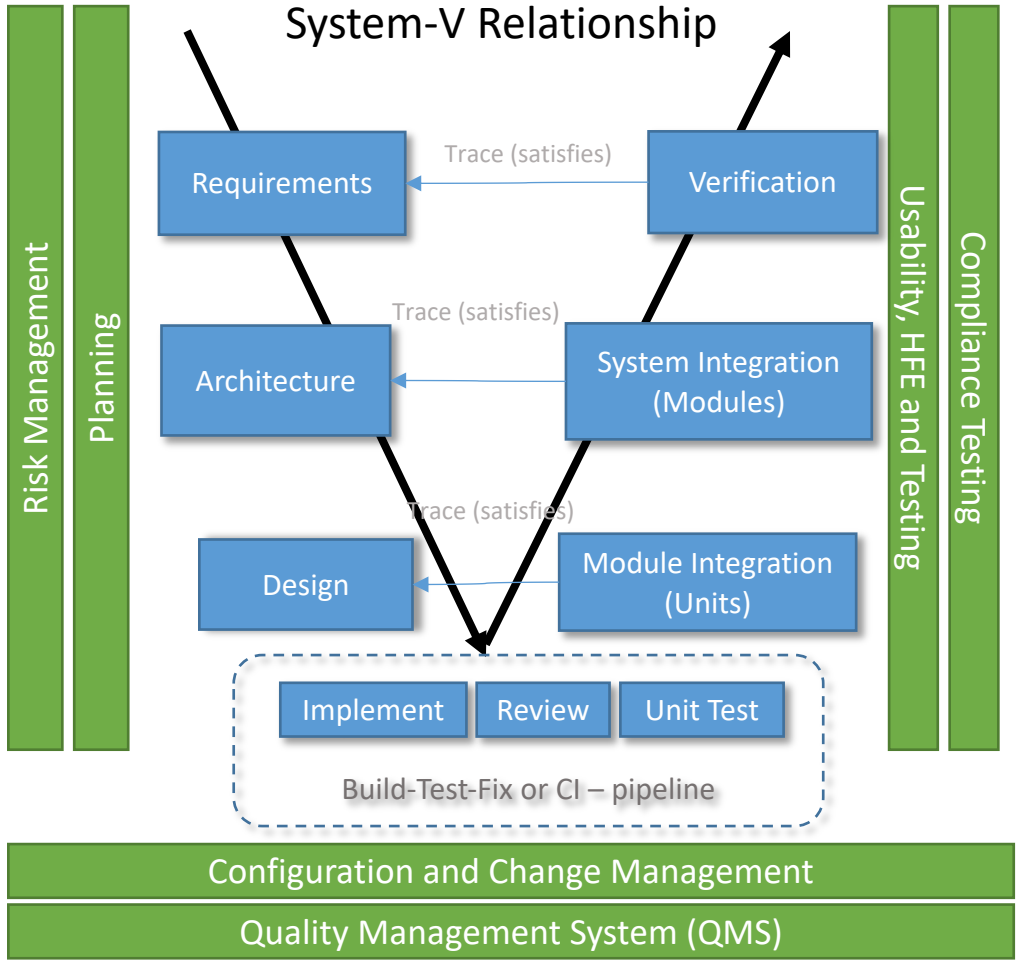
IEC 62304 –Adopting and Extending Development Process

5.8 - Release

- Execute an approval process per plan (5.1)
- Execute a release process per plan (5.1)
- Ensure all ACTIVITIES and TASKS per plan (5.1) are completed and documented (
- Includes any residual ANOMALIES
- Consider process replication, labeling, packaging, storage and transport (delivery)

The approval process creates a **contract**, acknowledging that a release is ready and available for use officially.

The stakeholders here should primarily be Project Management, Systems and Technical Leads but also can be Manufacturing or external contributors



IEC 62304 –Adopting and Extending Development Process

5.8 – Release

Document version information and archival through the configuration and change management processes (Part 6, 8 and 9)

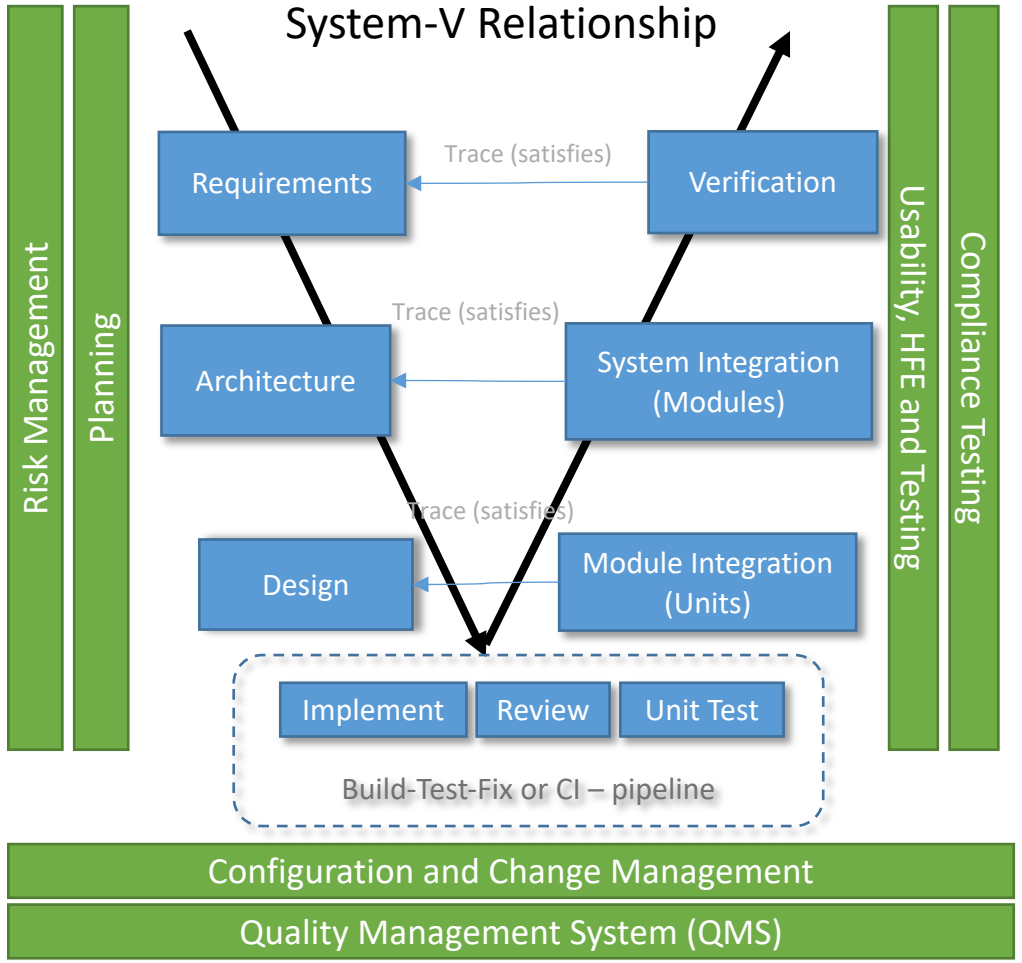
The release of artifacts should be captured in an index for documents or a compatibility table for objects

Most common version formats –

Semantic numbering: 3 dot technique – major.minor.build (patch)

Alphanumeric: Rev A or Rev 1.

System	Mechanical Assembly	Electrical PCBA	Firmware	Application
V0.6	REV B	REV D	V0.9.3	Usability (Formative)
V0.7	REV C	REV D	V0.9.5	EMI/EMC
V0.9	REV C	REV E	V1.0.1	System Verification
V1.0	REV D	REV E	V1.2.3	Commercial



Summary

- Adopting and extending IEC 62304 principles beyond software provides for a consistent product life cycle, specifically by aligning the development life cycles
- Minimize or eliminate confusion around terms and definitions
- Expose potential project risk and opportunities to improving the project planning by improving work breakdown structures
- Breaks down “siloes”, encourages collaboration from requirement through architecture, design and reviews of implementation.
- Provides additional guard-rails for a disciplined, requirements and risk-based development effort.

Questions!

Thank You!