# A SYSTEM DESIGN CONTROL PROCESS FOR MEDICAL DEVICE SOFTWARE DEVELOPMENT

**Joseph Akyeampong**
(Sr Systems Engineer)

Medtronic

The information presented herein is the personal view of the presenter and not to be taken as an official Medtronic position or endorsement.

# AGENDA

- **Purpose**
- **Background**
  - ❑ Medical Device SW – What Is It?
  - ❑ Medical Device Technology Refresh
- **FDA Design Controls (21CFR 820.30)**
- **System Design Controls Process (SDCP)**
  - ❑ Design Inputs
  - ❑ Design Outputs
  - ❑ Design Verification
  - ❑ Design Validation
- **Medtronic Medical Device Software Development Projects**
- **Q&A**

# PURPOSE

- Define Medical Device Software & Its Significance
  - ❑ *What are the drivers of medical device software development?*
- Describe FDA Design Controls
- Present a Systems Engineering process model (SDCP) for Medical Device Software Development
  - ❑ *Why is it needed?*
  - ❑ *How does it help reduce cost and improve quality?*
  - ❑ *Provide examples of how it has been successfully implemented at Medtronic*

# BACKGROUND

## Medical Device Software  - What Is It?

Software intended to be used for one or more medical purposes that perform these purposes without being part of a hardware medical device  [FDA].

Medical device software is used across a broad range of technology platforms

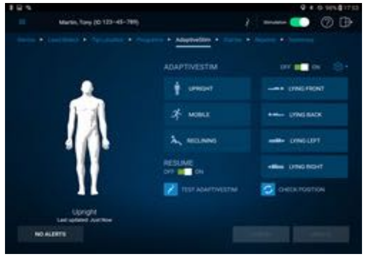
- ❑ Medical device platforms (custom-built)
- ❑ Commercial Off The Shelf (COTS) platforms (e.g. tablets, smart phones, laptops, PCs etc.)
- ❑ Virtual networks (i.e. cloud)

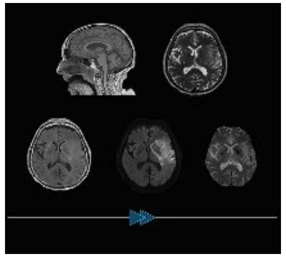
Other names: software as a medical device (SaMD) standalone software, medical device software, health software

# BACKGROUND

## Examples of Medical Device Software

- ❑ Software that controls a medical device e.g. an implantable neurostimulator (pain/brain), insulin pump or pacemaker
- ❑ Software that performs imaging and diagnostic procedures e.g. MRI
- ❑ Software that controls inflation and deflation of a blood pressure cuff through a mobile platform
- ❑ Software that uses the digital camera of medical scopes to diagnose a condition
- ❑ Treatment planning applications that supply information
- ❑ BMI and body fat calculators, and heart rate monitors





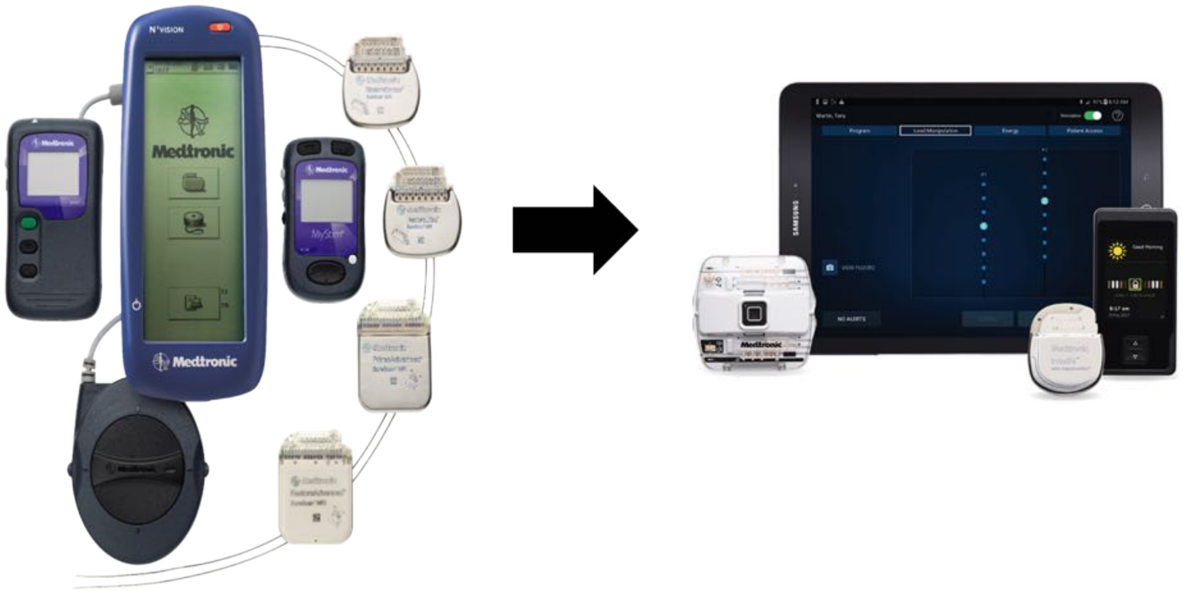
# BACKGROUND

## Significance of Medical Device Software

- ❖ Provides the interface for controlling and monitoring medical devices
- ❖ Scalability – allowing expansion of the functional capabilities of medical devices
  - ❑ Bodily interaction (i.e. deliver stimulation or a drug)
  - ❑ Monitoring and control of therapy
  - ❑ Wireless integration
  - ❑ System security
- ❖ Device/Platform independent – i.e. custom, commercial off-the-shelf, cloud, Windows, iOS, Android etc.
- ❖ Accessibility – i.e., downloadable or pushed apps
- ❖ Efficient and effective diagnoses and treatment of disease conditions

# BACKGROUND

## Medical Device Technology Refresh



### Drivers

- ❖ Accessible healthcare
- ❖ Personalized treatment
- ❖ Advancements in electronics

### Benefits

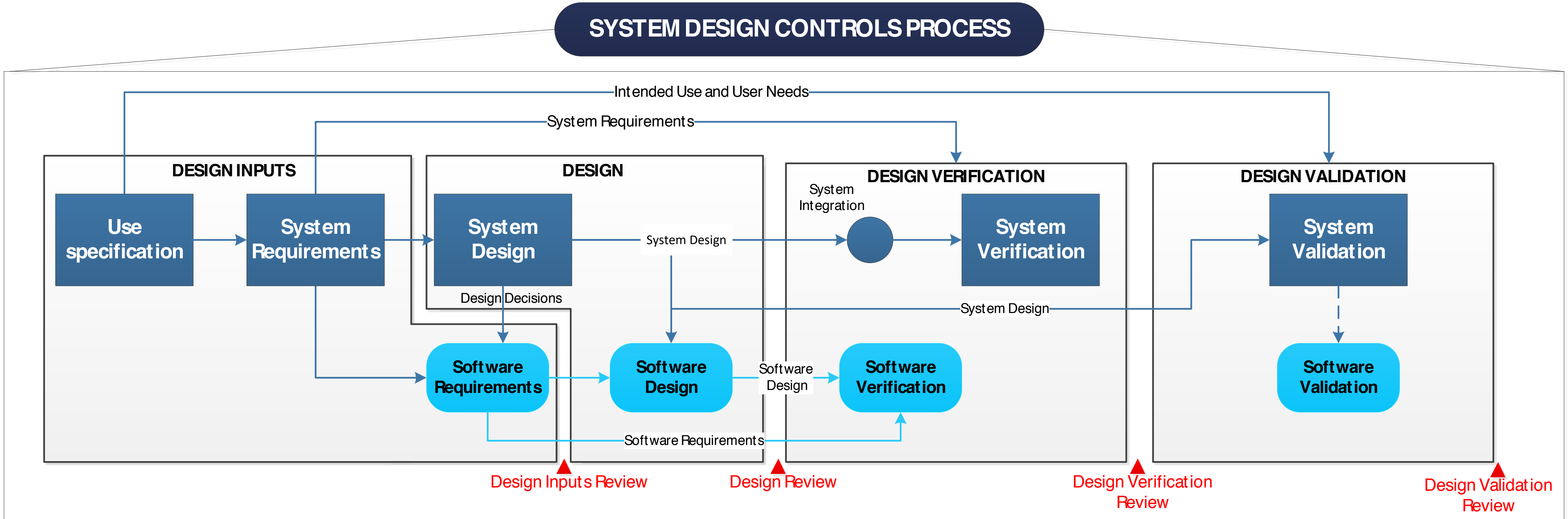
- ❖ Competitive advantage
- ❖ Reduced cost
- ❖ Familiarity/ adoption
- ❖ Improved user experience

# SYSTEMS DESIGN CONTROL PROCESS (SDCP)

## Why SDCP?

- ❖ Establishes a system-driven process for medical device software development
- ❖ Identifies the sequence of System and Software development activities for effective/efficient product development
- ❖ Emphasizes the need for review at the end of each design control phase to ensure quality outputs
- ❖ Built on a foundation of risk management to ensure safety and efficacy of the medical device software

# SYSTEMS DESIGN CONTROL PROCESS (SDCP)



SYSTEM DESIGN CONTROLS PROCESS

Intended Use and User Needs

System Requirements

DESIGN INPUTS

Use specification

System Requirements

DESIGN

System Design

Design Decisions

System Design

Software Requirements

Software Design

Software Requirements

DESIGN VERIFICATION

System Integration

System Verification

System Design

Software Design

Software Verification

DESIGN VALIDATION

System Validation

Software Validation

Design Inputs Review

Design Review

Design Verification Review

Design Validation Review

1. Identify design input sources
2. Create use specification
3. Define system and software requirements /establish traceability

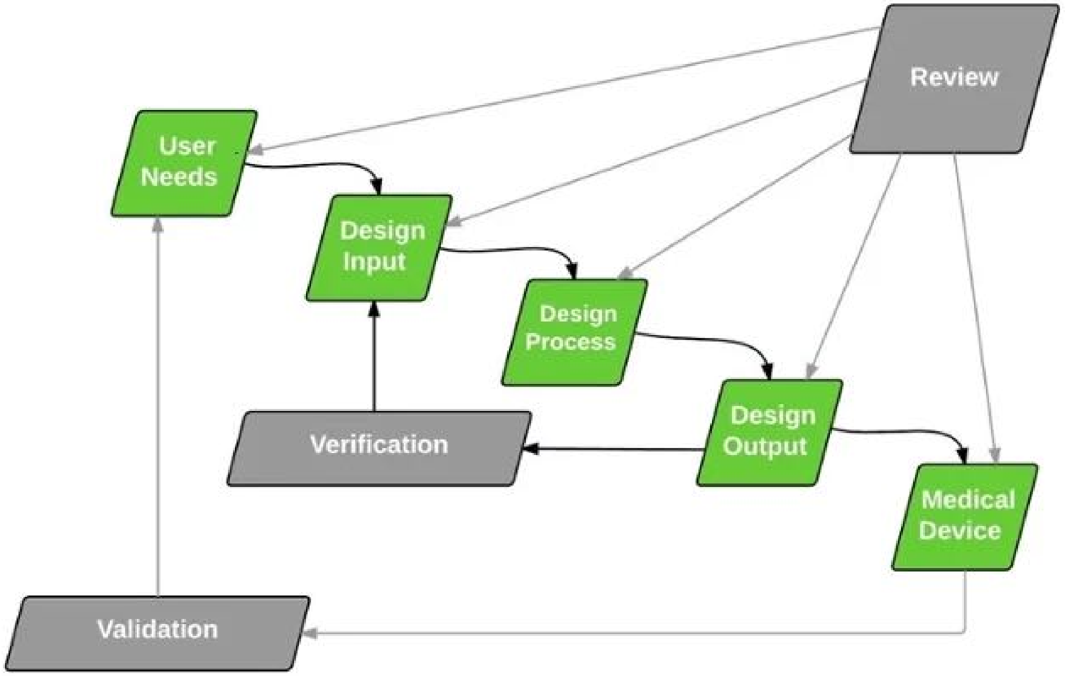
1. Create system/software architectures
2. Develop system design
3. Perform risk analyses
4. Develop software design

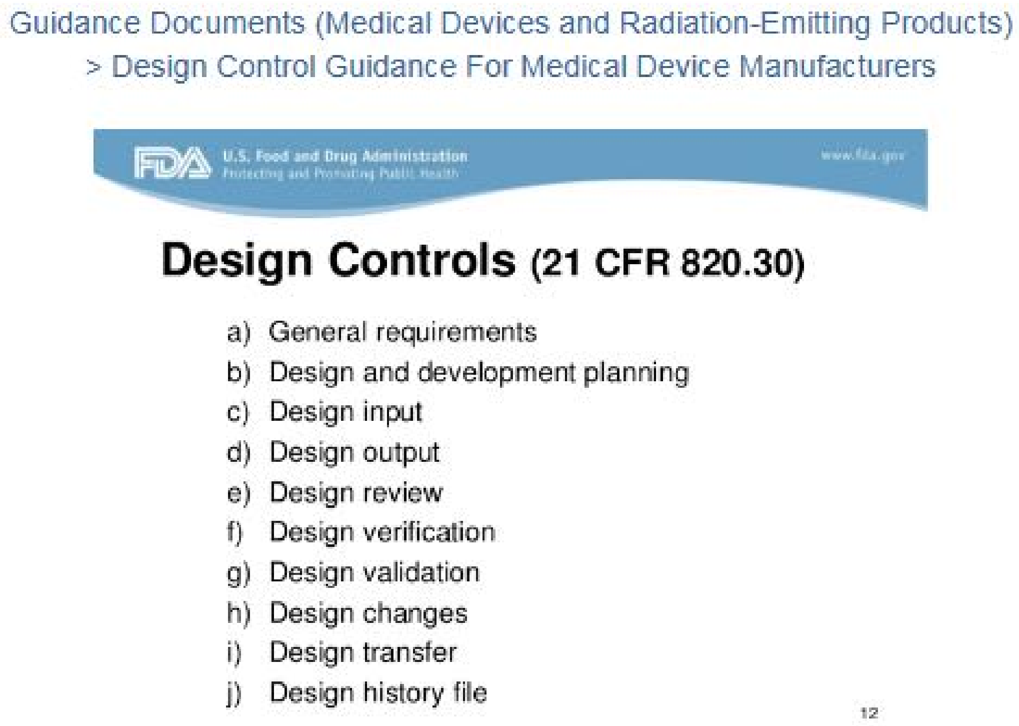
1. Create system and software design verification plan/protocol
2. Complete SW design verification
3. Complete system design verification
4. Create system and software design verification reports

1. Create design validation plan/ protocol
2. Complete design validation
3. Create design validation report

Quality System / Risk Management

# FDA DESIGN CONTROLS(21 CFR 820.30)





Guidance Documents (Medical Devices and Radiation-Emitting Products)
> Design Control Guidance For Medical Device Manufacturers

FDA U.S. Food and Drug Administration
Protecting and Promoting Public Health
www.fda.gov

## Design Controls (21 CFR 820.30)

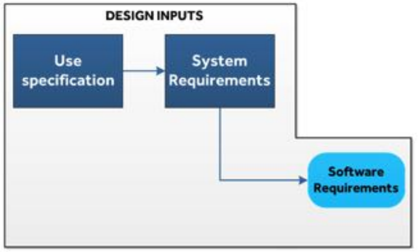
a) General requirements
b) Design and development planning
c) Design input
d) Design output
e) Design review
f) Design verification
g) Design validation
h) Design changes
i) Design transfer
j) Design history file

12

INCOSE Healthcare Working Group

4th Annual Systems Engineering in Healthcare Conference

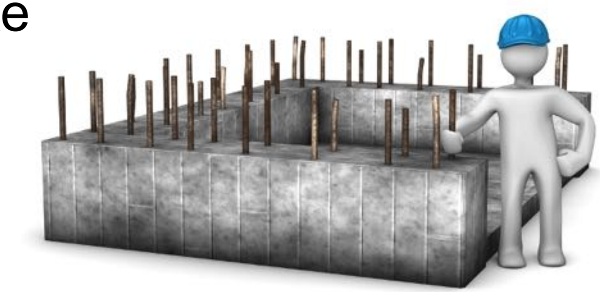
# DESIGN INPUTS



## Overview

- Design inputs (DI) establish the foundation for medical device product development

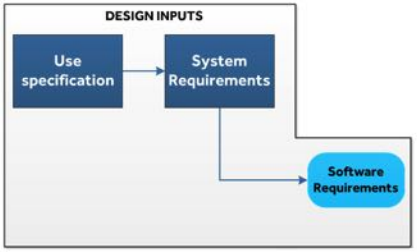
- Defining DIs can be time-consuming
  - ❑ Requires a disciplined approach to identifying appropriate design inputs



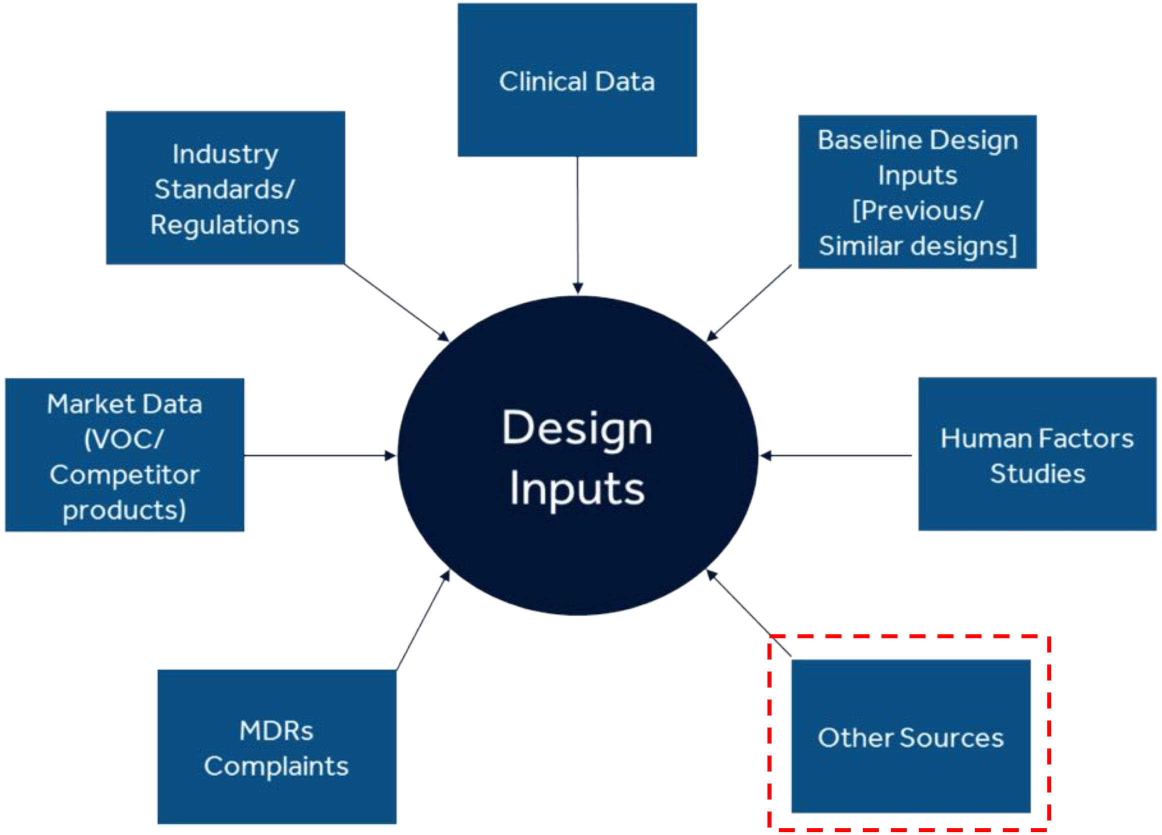
## Goals For Defining Design Inputs

- Appropriately capturing all user and stakeholder needs
- Adequately capturing applicable requirement types – (functional, performance, usability, regulatory etc.)
- Ensuring DIs are clear, unambiguous, non-conflicting, verifiable, "validatable"

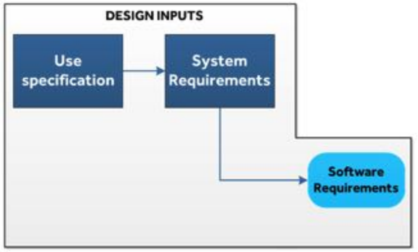
# DESIGN INPUTS



## Identify Design Input Sources


Clinical Data
Industry Standards/ Regulations
Baseline Design Inputs [Previous/ Similar designs]
Market Data (VOC/ Competitor products)
Design Inputs
Human Factors Studies
MDRs Complaints
Other Sources

# DESIGN INPUTS



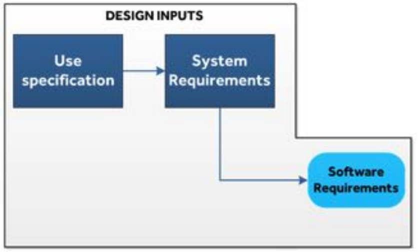
## Create Use Specification (IEC 62366-1)

### Elements of the Use Specification

- □ intended use
- □ indications of use
- □ user profiles
- □ environment profiles
- □ user needs
- □ use cases/scenarios

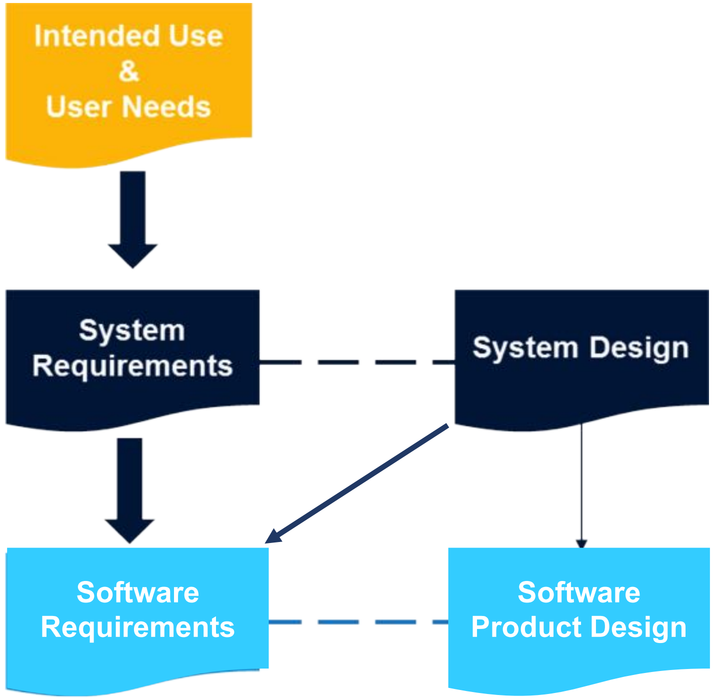
| Question | Answer |
|---|---|
| What will the medical device be used for? | Intended Use |
| Which medical conditions will the medical device diagnose, treat, prevent or mitigate? | Indication of Use |
| Who will use the medical device? | User Profiles |
| Where will it be used? | Environment Profiles |
| Why is the medical device essential to the user? | User Needs |
| How will the user interact with the medical device? | Use Cases/Scenarios |

# DESIGN INPUTS

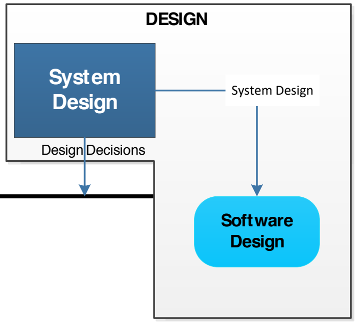


## Define Requirements / Establish Traceability

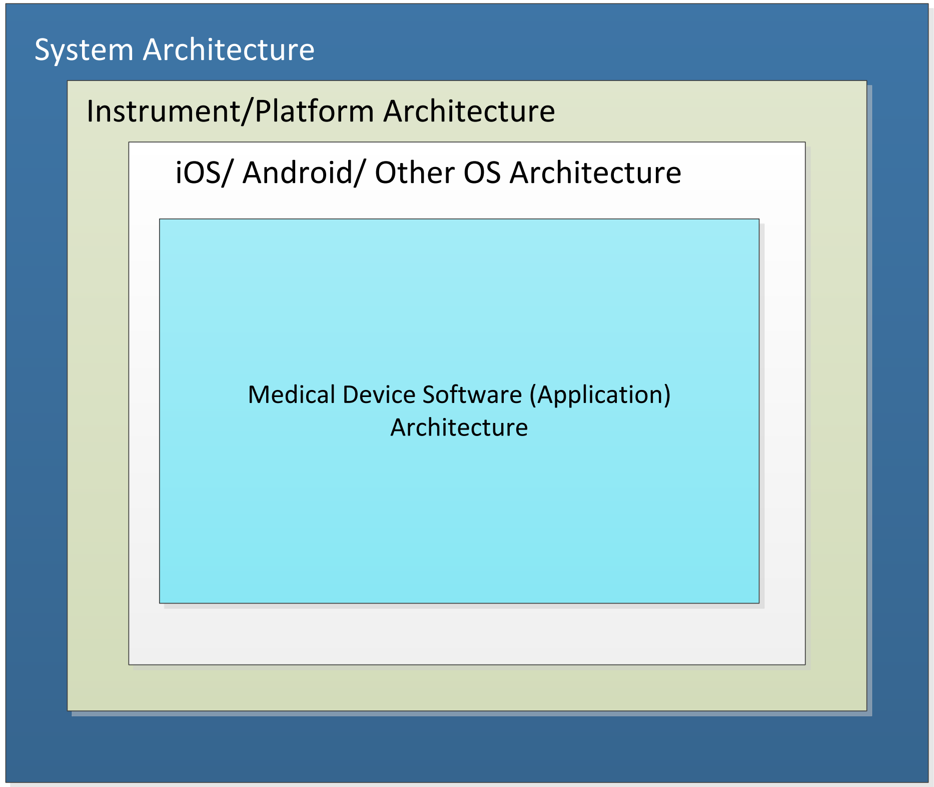
- ❖ System Requirements
  - ❑ Functional/performance
  - ❑ Operational (including Security)
  - ❑ Environmental
  - ❑ Usability
- ❖ Software Requirements
  - ❑ Functional – Capabilities to configure/control medical device
  - ❑ Informational - Capabilities to enter, update or view information about the medical device



# DESIGN

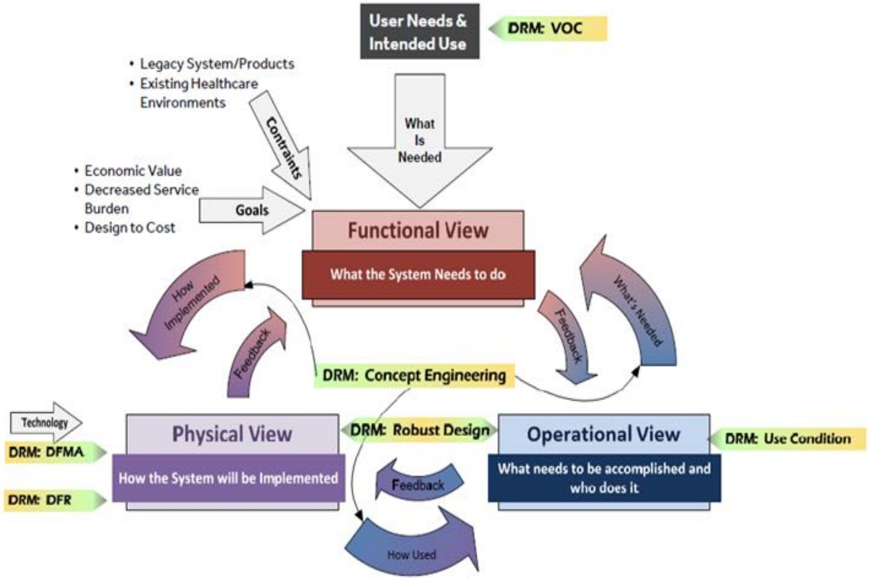


## Define Architectures > System Architecture

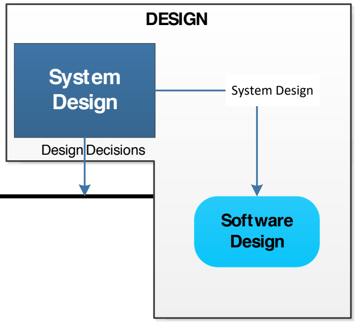

System Architecture
Instrument/Platform Architecture
iOS/ Android/ Other OS Architecture
Medical Device Software (Application) Architecture

### System Architecture Views

- ❑ Functional
- ❑ Physical
- ❑ Operational



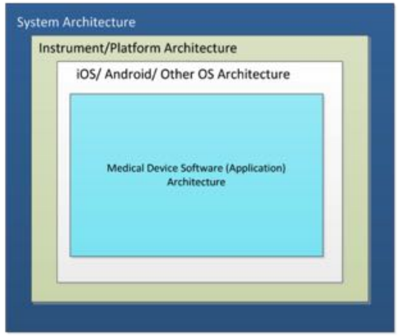
# DESIGN



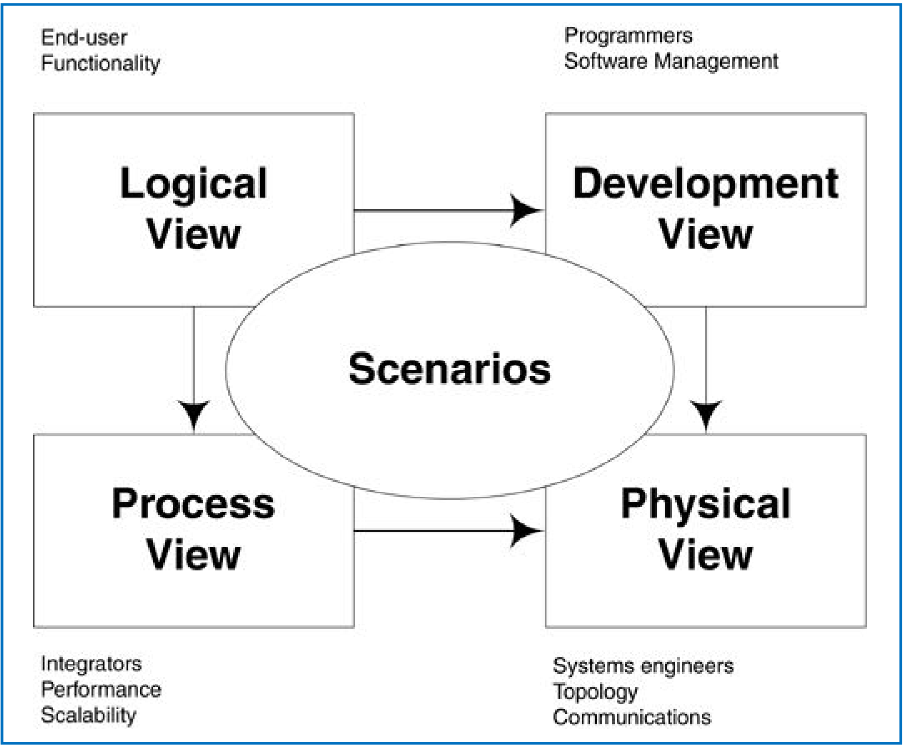
## Define Architectures – Software Architecture

### Software Architecture Views

- ❑ Logical
- ❑ Development
- ❑ Physical
- ❑ Process
- ❑ Operational (Scenarios)*


System Architecture
Instrument/Platform Architecture
iOS/ Android/ Other OS Architecture
Medical Device Software (Application) Architecture

System Architecture

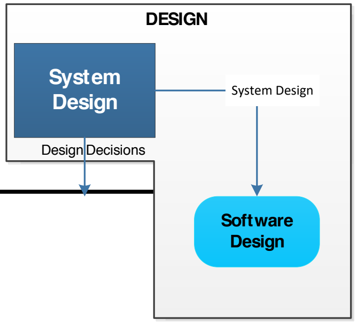


**Krutchen 4+1 Software Architectural View Model**

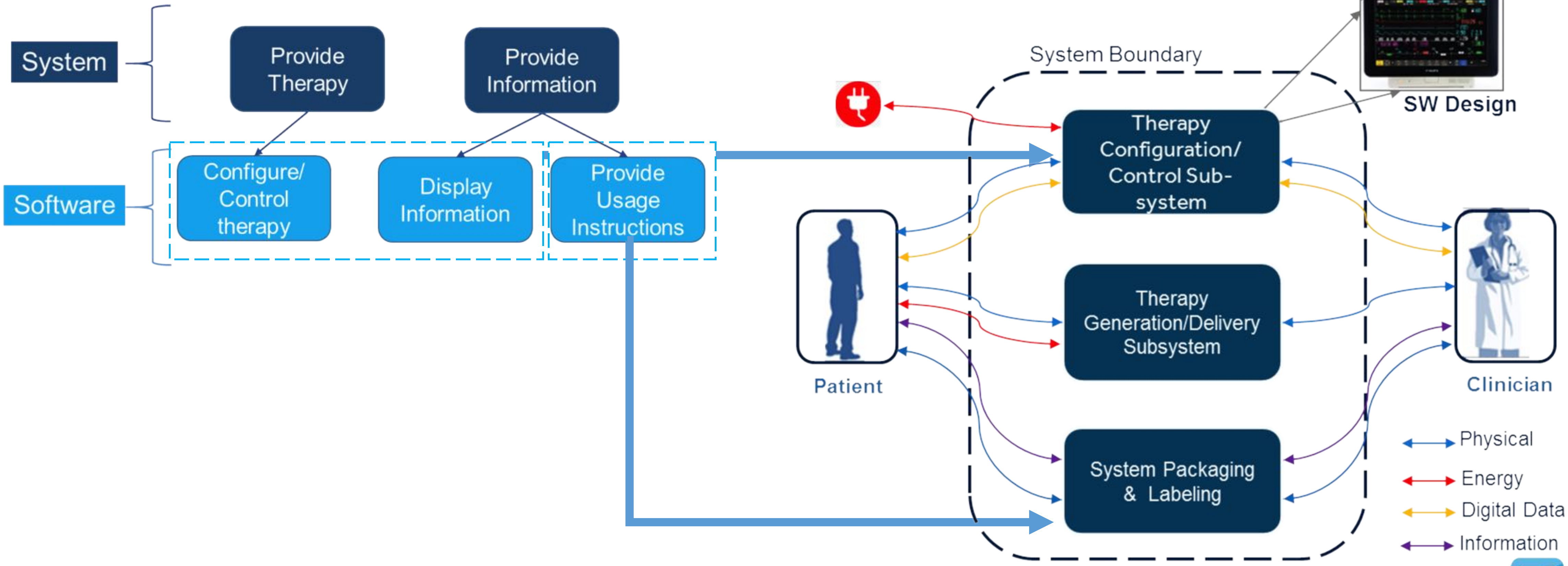
INCOSE Healthcare Working Group

4th Annual Systems Engineering in Healthcare Conference

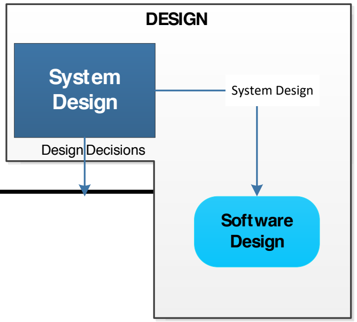
# DESIGN



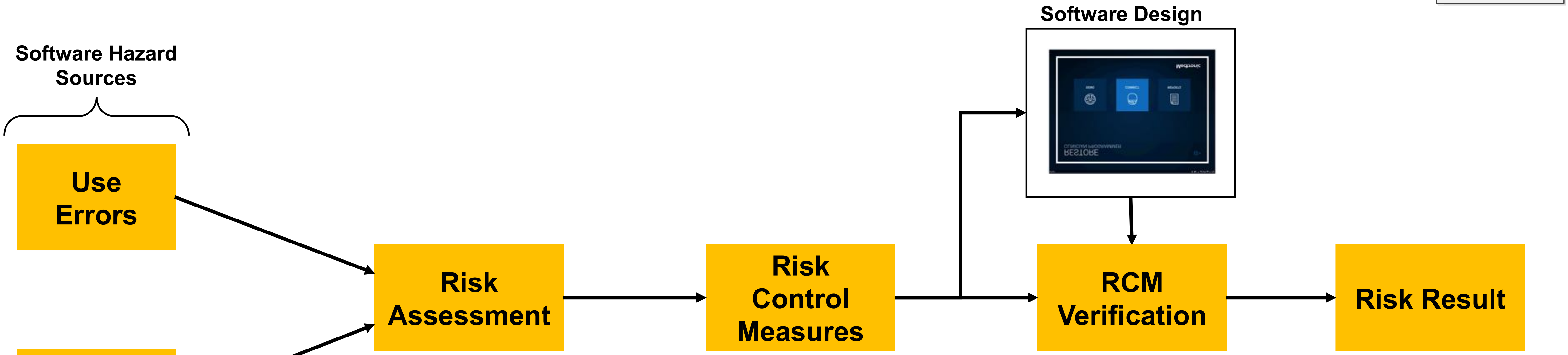
## System Design


Define Functions
Allocate Function to Physical Elements
System
Software
Provide Therapy
Provide Information
Configure/ Control therapy
Display Information
Provide Usage Instructions
System Boundary
SW Design
Therapy Configuration/ Control Sub-system
Therapy Generation/Delivery Subsystem
System Packaging & Labeling
Patient
Clinician
Physical
Energy
Digital Data
Information

# DESIGN



## Perform Risk Analysis



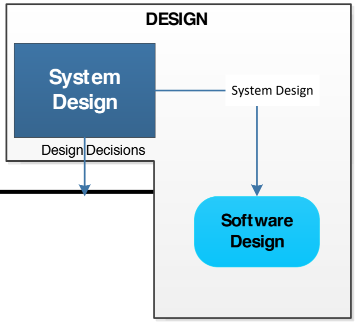
- Define Sequence of Events
- Identify Hazards
- Hazardous Situations
- Determine Severity
- Determine Occurrence

- Inherent Safety by Design
- Protective Measures
- Detection and notification
- Labeling and training

- Software Output vs. RCM

- Evaluate Residual Risk
  * Goal: ALAP

# DESIGN



## Software Design

**System Functions Allocated to Software**

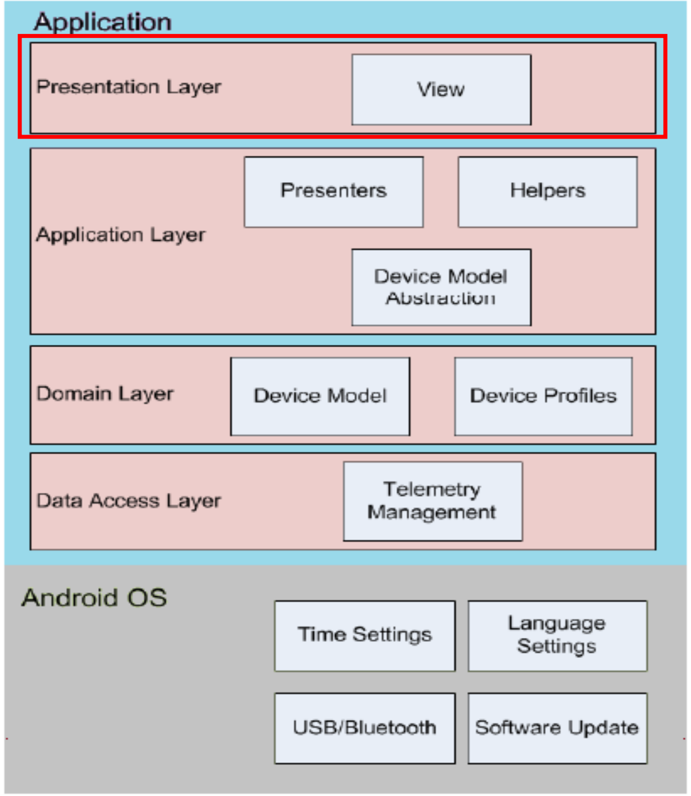
**Provide therapy**

- ❑ Configure/Control therapy
- ❑ Provide Information

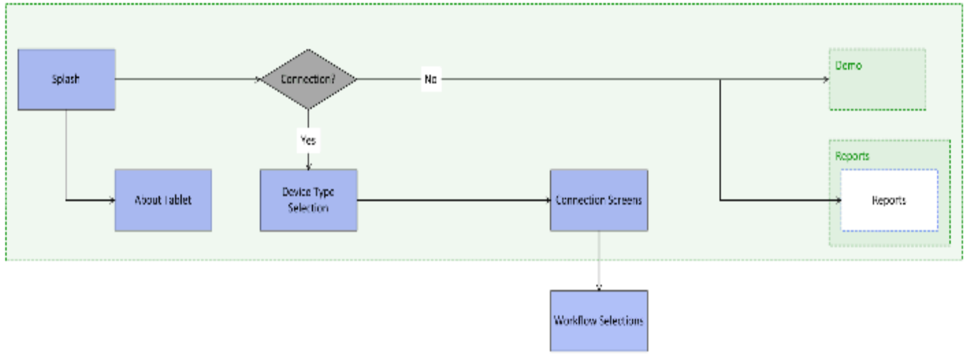
**Provide information**

- ❑ Display information
- ❑ Display instructions

### Software Logical View

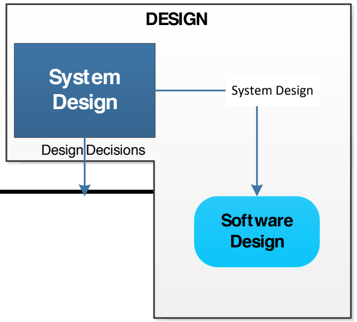


### 1. Create Site Map

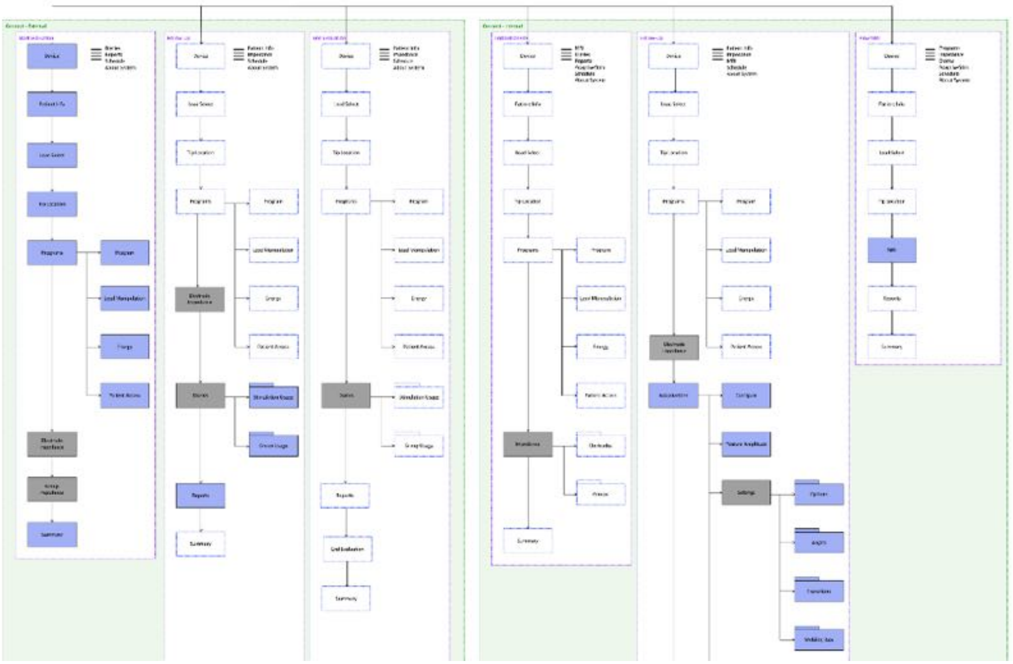


# DESIGN

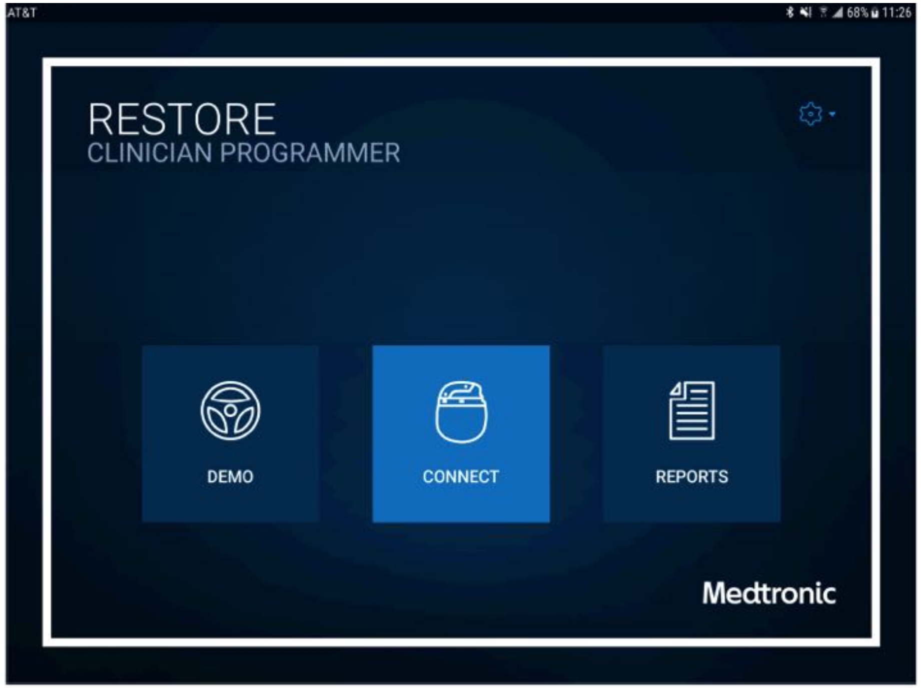
## Software Design



### 2. Create Workflows



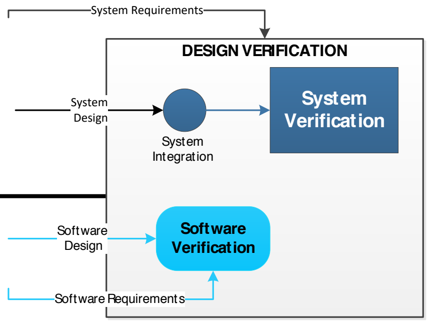
### 3. Develop Screen Designs



INCOSE Healthcare Working Group

4th Annual Systems Engineering in Healthcare Conference

# DESIGN VERIFICATION



## System Integration and Testing

### System Integration

❖ Connect system components to assess functionality

### System Integration Testing

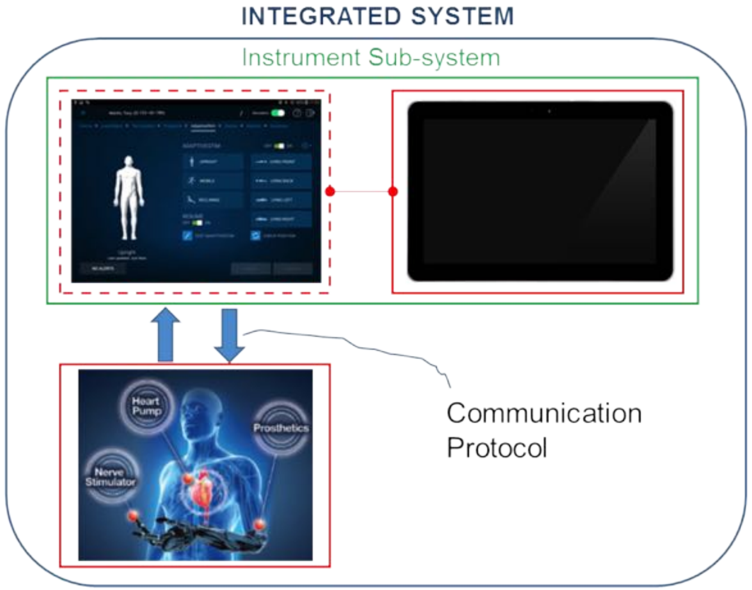
❖ Test of the integrated system

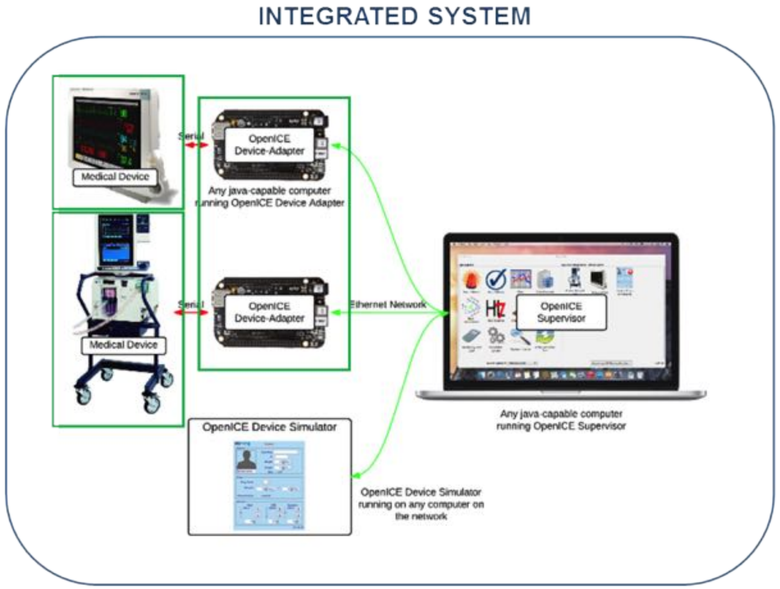
❖ Identify issues

❖ Fix the issues

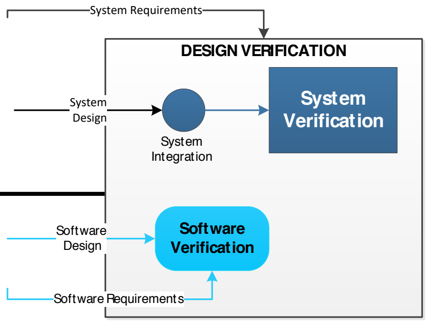
Examples:

- ❑ Functionality testing
- ❑ Use case testing
- ❑ Compatibility testing
- ❑ Free-form testing

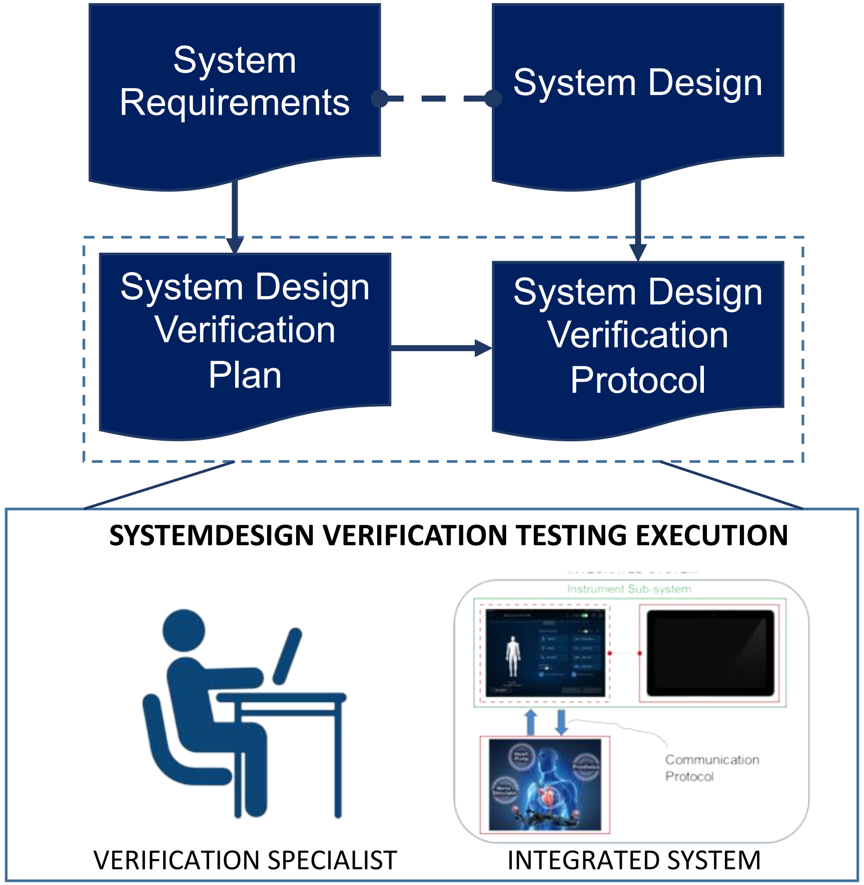



INTEGRATED SYSTEM
Medical Device
Serial
OpenICE Device-Adapter
Any java-capable computer running OpenICE Device Adapter
Medical Device
Serial
OpenICE Device-Adapter
Ethernet Network
OpenICE Supervisor
Any java-capable computer running OpenICE Supervisor
OpenICE Device Simulator
OpenICE Device Simulator running on any computer on the network

# DESIGN VERIFICATION



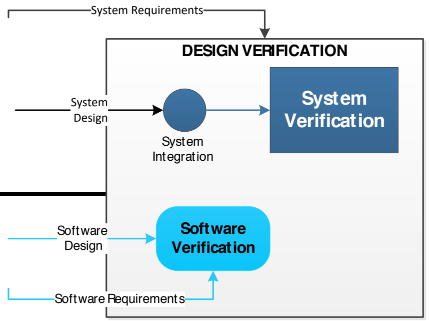
## System Design Verification



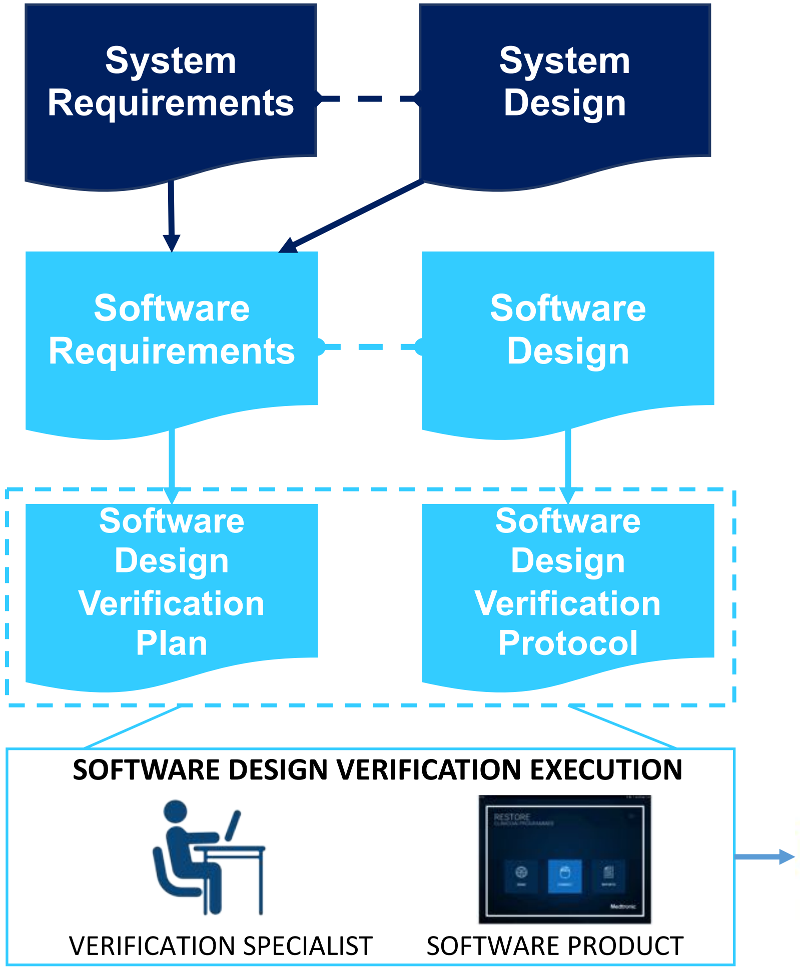
- System Design Verification
  - ❑ Verifies system requirements
  - ❑ Did we build the system right?

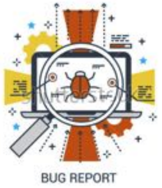
- System Design Verification Techniques
  - ❑ Test
  - ❑ Inspection
  - ❑ Demonstration
  - ❑ Leverage Child Verification
  - ❑ Analysis – Similarity/ Rationale

# DESIGN VERIFICATION



## Software Design Verification


System Requirements
System Design
Software Requirements
Software Design
Software Design Verification Plan
Software Design Verification Protocol
SOFTWARE DESIGN VERIFICATION EXECUTION
VERIFICATION SPECIALIST
SOFTWARE PRODUCT

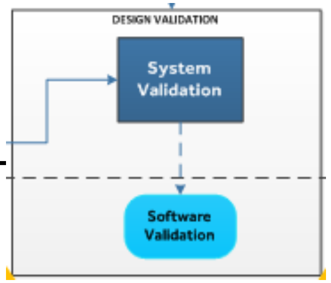


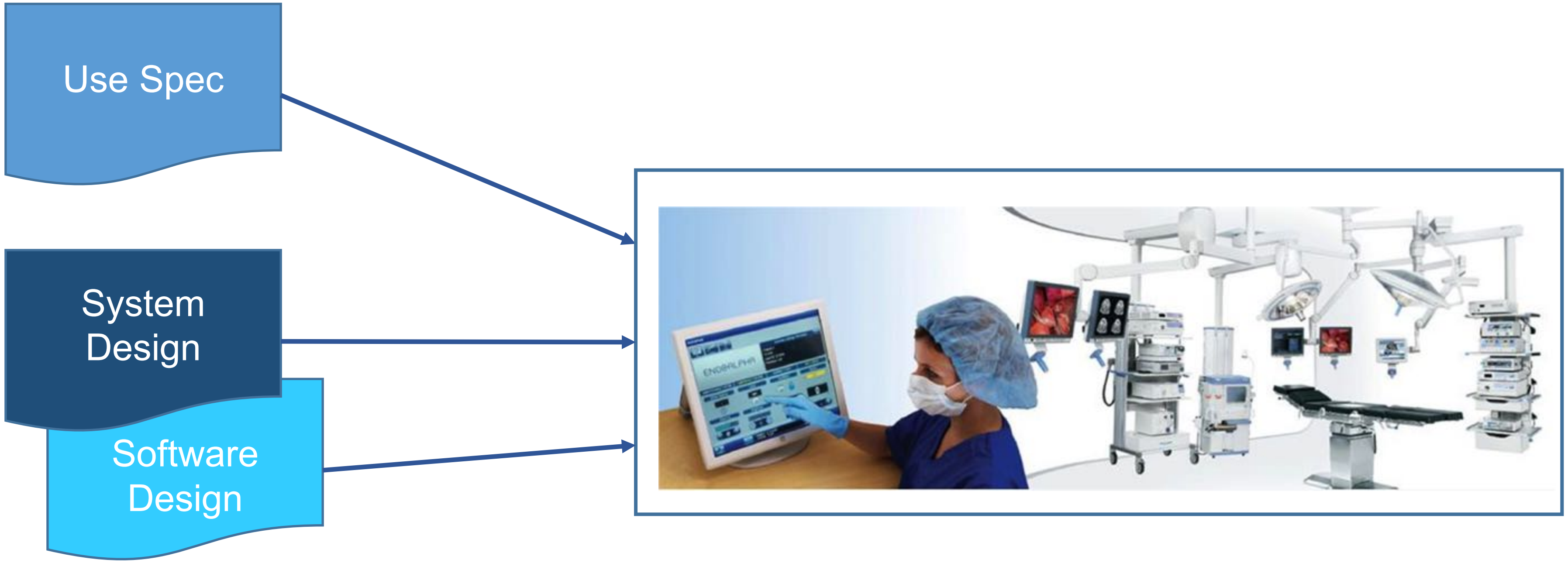
- Software Design Verification
  - ❑ Verifies software requirements
  - ❑ Did we build the product right?

- Software Design Verification Techniques
  - ❑ Feature Acceptance Tests
  - ❑ User Story Acceptance Tests
  - ❑ Software System Tests
  - ❑ Integration Tests

# DESIGN VALIDATION

*Did we build the right product?*





**Validation Methods**

Human Factors Summative Tests

System Validation Test

Analysis Rationale/ Similarity

Leverage Verification

- ❖ Validate the final design against the intended use /user needs in actual or simulated use environments
- ❖ Focus on features with high risk (harm severity) as identified through risk analyses
- ❖ Focus on evaluating usability of software

Thank you

Q & A