MEDICAL DEVICE ARCHITECTURAL SPECIFICATION

Engagement ID :

Engagement Name :

Client Name :

COMPANY CONFIDENTIAL · SEC 1 · Capgemini Document

Company Confidential - Is all regular business owned and created by Capgemini employees which is not intended to be publicly available and only intended to be used by Capgemini internally.

Table of Contents

Company Confidential. © 2023 Capgemini. All rights reserved.

Work Instructions:	
1.	All write up in **PINK**, are GENERIC EXAMPLES to guide in preparing a project write-ups for that section. These need to be converted to 'Normal' style for project write-up and modified or deleted as per testing objective and scope at hand.
2.	All write-ups in <<blue within a greater than and less than sign>>, are write-up that may need to be deleted or fill with information as per instruction provided inside and color provided.
3.	All write-ups in 'Normal' Style (see below), are common write-up and may need minor project-specific changes.
4.	The write-up in **BLUE** are WORK INSTRUCTIONS/ GUIDANCE and need to be completely removed from the following the instructions.
5.	Regenerate the Table of contents, after completion
6.	Fonts used:
	Normal: Ubuntu MS 10 (Color Automatic)
	Normal Bold: Ubuntu MS 10 Bold (Color Dark blue)
7.	Delete this box after completion

Company Confidential. © 2023 Capgemini. All rights reserved.

1. INTRODUCTION

1.1 PURPOSE

The Medical device architecture Specification (MDAS) describes the detailed architecture specification for the << Medical Device / Sub-System name and model >>.
This document will be referred by design and verification team to create / develop the detailed component / unit design. It includes architecture covering Hardware, Firmware, Software and Mechanical design.

1.2 SCOPE

This document focuses on the System level architectural specifications of the << Medical Device / Sub-System name and model >>. The <<name the methodology>> in QMS > Delivery process has been used for the design and development lifecycle of medical device .These specifications are derived from the << Medical Device Requirements Specifications (MDRS) or Product Requirement Document (PRD)>> document.
The Scope includes:

- A general description of the system / sub-system.
- The logical architecture of software, the layers and top-level components.
    - The physical architecture of the hardware (Electrical and Mechanical) on which the software runs.
- *The justification of technical choices made.*

1.3 INTENDED AUDIENCE

<<Insert the members who are required to be involved in the design/ testing of the product as well as the client if involved. >>
This document, henceforth, should be used as direct reference by the members of the organization project team that designs and implements (hardware design, schematics and PCB layout, software design, unit design, coding, Mechanical design) system / sub-system requirements, and by the team that tests the system / sub-system.
It is assumed that the person reading this document has gone through and understood the Medical Device Requirements Specifications (MDRS) or Product Requirement Document (PRD).

1.4. Definitions And Acronyms

1.4.1. Definitions

<<Add here abbreviations. >>

Word	Definition
<Add the word>	<Add corresponding meaning>
Unit	Any entity that cannot be sub divided further.
component	Group of units assembled together to form component. This is also referred to as module.
Sub-System	Group of components integrated to form a part of a system
System	Collection of sub system (if applicable) or components integrated together to achieve the defined intended use (Medical function). This is the end product.

1.4.2. Acronyms

<<Add here words definitions. >>

Company Confidential. © 2023 Capgemini. All rights reserved.

Acronyms	Expansion
<Add acronym>	<Add corresponding expansion>
FDA	Food And Drug Administration
WEEE	Waste Electrical And Electronic Equipment
CRC	Cyclic redundancy check

1.5. References

1.5.1. Document References

<<Add here document references. >>

Sr. No;	Document Number / Identifier	Document Title
	<Add Document Number >	<Add your documents references with version. One line per document>
1	AAA/BBB/CCCC	Guideline document for testing V 2.0
2	AAA/BBB/DDDD	Test Strategy
3	AAA/BBB/DDDD	Medical Device Requirement Specification

1.5.2. Standard and regulatory References

<<Add here standard references. >>

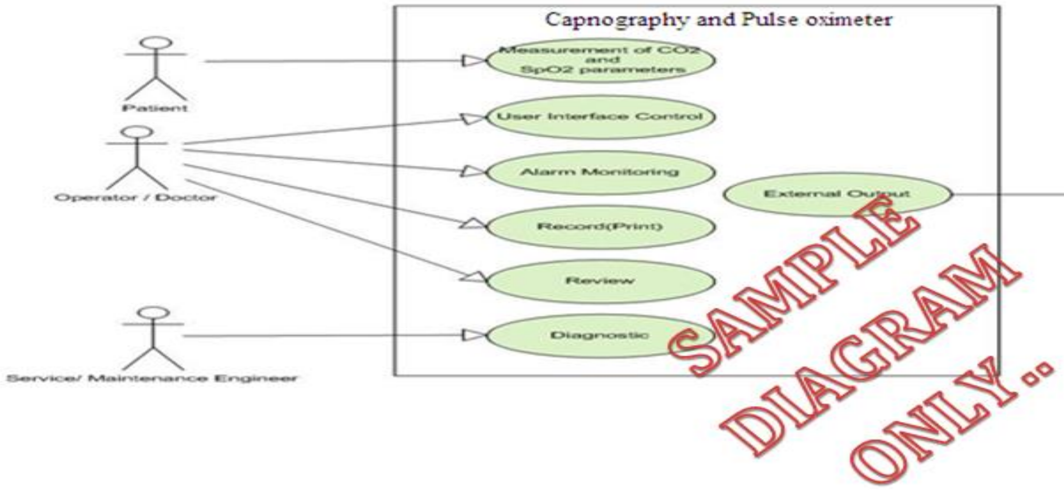
Sr. No;	Document Number / Identifier	Document Title
	<Add standard references number>	<Add your standard references with version number. One line per document>
1	IEC 60601-1	Basic safety and essential performance of medical devices Edition 3.0
2	ISO 13485	Medical devices - Quality management systems - Requirements for regulatory purposes.

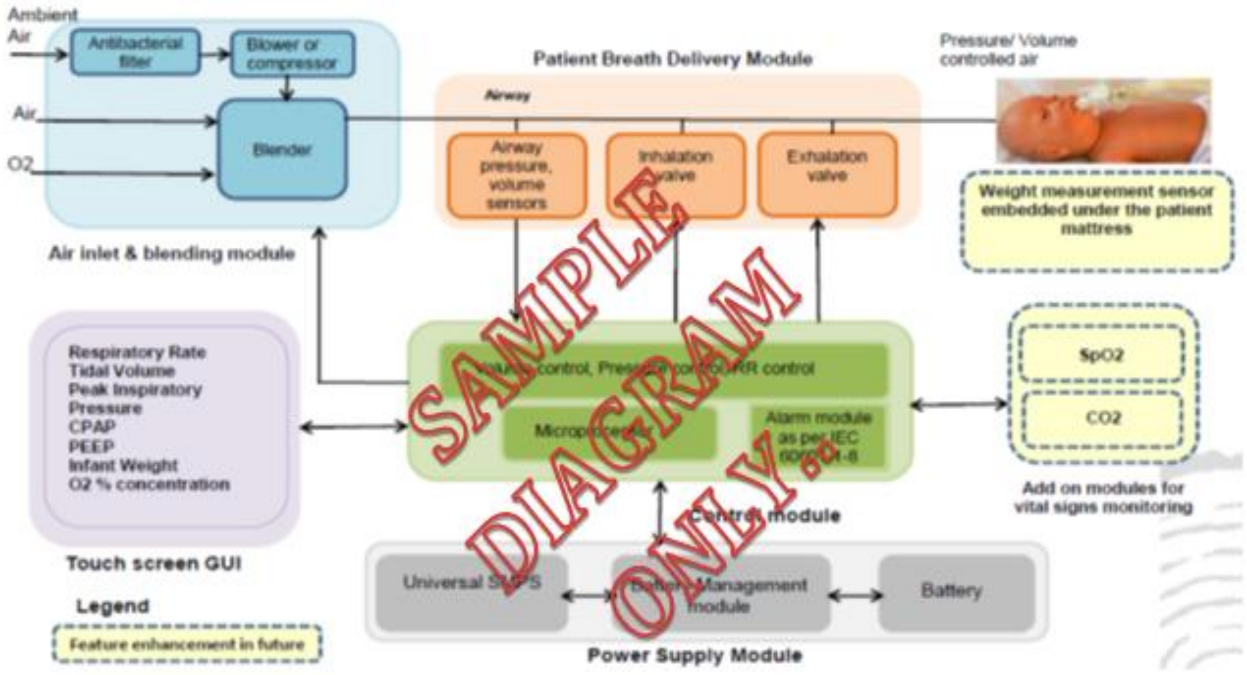
Company Confidential. © 2023 Capgemini. All rights reserved.

2. Architecture

2.1. Architecture overview

<<Give a general description of the system along with block diagram as defined below. [**Note**: Image below is just for representation and should not be considered as standard reference drawing.]



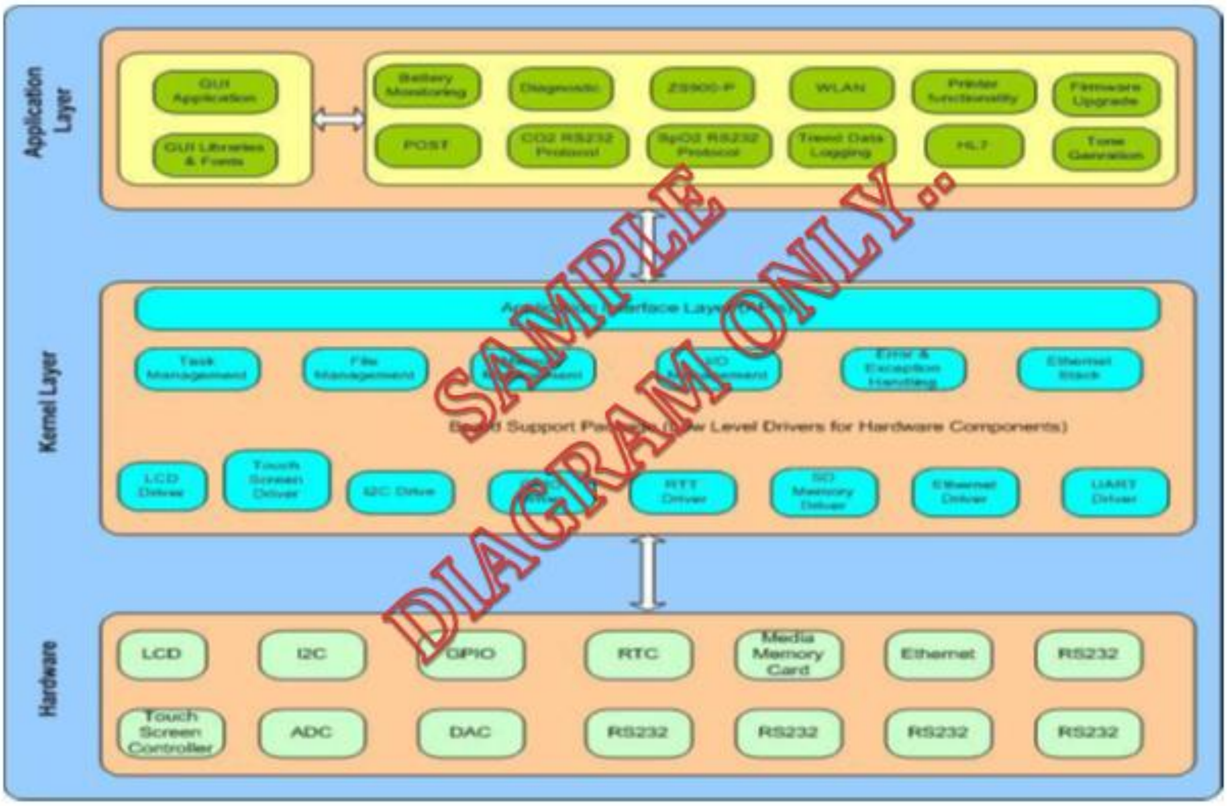

Ambient Air
Antibacterial filter
Blower or compressor
Air
O2
Blender
Air inlet & blending module
Patient Breath Delivery Module
Airway
Airway pressure, volume sensors
Inhalation valve
Exhalation valve
Pressure/ Volume controlled air
Weight measurement sensor embedded under the patient mattress
Respiratory Rate
Tidal Volume
Peak Inspiratory Pressure
CPAP
PEEP
Infant Weight
O2 % concentration
Touch screen GUI
Volume control, Pressure control, RR control
Microprocessor
Alarm module as per IEC 60601-1-8
Control module
SpO2
CO2
Add on modules for vital signs monitoring
Universal SMPS
Battery Management module
Battery
Power Supply Module
Legend
Feature enhancement in future
SAMPLE DIAGRAM ONLY..

- Product name /model/ variants (as applicable)
- In what environment medical device will be used / operated (e. g. home, near patient bed, operating room etc.…)
- Who the users are (e. g. person under treatment, operator)
- What application it is for (e.g. Hematology, Neurology, Cardiology etc.),
- The main functions (performance and intended use related)
- The main interfaces, inputs and outputs.
- Features:
  - Power supply mode(Internal / External or Both and class of power supply),
  - Class 1 or Class 2 medical device,
  - Nature of operation (Continuous or non-continuous operation)
  - Applied parts and their nature (Type B, BF, CF, etc.)

  - Brief description on each parameter to measure as well as to outputs.
  - Communication features (i.e., Interface, wired/wireless),
  - Outputs on product, (E.g.: Display, thermal printer)
  - Memory storage features,
  - External module/ interfaces/device connections. (E.g.; connection to Network, connection to Central Nursing Station, PC, Printer etc.)
- And other key features with respect to the product (E.g. Interpretation of signals through software, verbal alarms, provision to connect specified external DC supply source such as vehicle battery, mounting provision, Ingress Protection (IPXX) etc.)
- Use or processing of Medicinal product (drug) (Describe the medicinal product and application in the use of this device. >>

2.2. Logical architecture overview

<<Note: Image below is just for representation and should not be considered as standard reference drawing. >>



<<Description of top level software (both embedded and application wise which would be later integrated for attaining the end application.), its components and their interactions/relationships are to be addressed here. Use UML package diagrams and/or layer diagrams and/or interface diagrams. Also describe the operating systems on which the software runs. **(Please define high level details about the individual s/w components themselves.)** The main purpose here is to gain a general understanding of how and why the software was decomposed, and how the individual parts/items/components work together to provide the desired functionality. **(Note:** If requirement (as stated by the PRD or MDRS) is achieved by hardware/ software/ combination of both together, they have to be mentioned under the corresponding "**hardware component description**" as well as "**software component description**" respectively.**)** >>

2.2.1. Software Component 1 description

<<Provide **brief** description about this software component. While preparing this description, the following parameters are to be considered (but not limited to) based on the design and requirement at hand;

- Top level component identification (e. g. Battery monitoring function, CO2 communication protocol, Diagnostic feature etc.)
- The purpose of the component. (e. g. In Battery monitoring function, to monitor battery charge status, control battery charging rate and mode, communicate status to main functional component etc.)
- Explanation of software structure within that component. Partitioning of functionality; (e. g. identification of critical, non-critical and supervisory sections of software in a system.), Diversity
- Its interfaces and interaction with other components, (i.e. between s/w- s/w and between s/w- h/w, internal & external)
- Its network interfaces, communication, security, protocols etc.
- Mention whether the Software component is to be purchased, re-used or built.
- The hardware resources it uses, (e. g. In Battery monitoring function use of separate controller and its features) Other examples that can be considered are, average RAM usage, peak RAM usage and peak frequency and duration, disk space for permanent data, disk space for cache data, average CPU usage, peak CPU usage and peak frequency and duration etc…
- Design Environment;(This should describe the software development methods (this can be briefly described and later explained in the corresponding plans), CASE tools, programming language, hardware and software designing platform, simulation tools, design and coding standards, compiling tools.).
- Reliability/redundancy of the software with respect to basic safety and essential performance.
- Defensive design; (e.g. limits on potentially hazardous effects by restricting the available Output power.)
- Device limitations due to software (e. g. Execution speed, response time, baud rate etc.)
- Internal software tests and checks (e. g. Self-test, Diagnostics mode etc.)
- Error and interrupt handling (e. g. Prioritization of error messages and critical tasks, error log)
- Fault detection, tolerance, and recovery characteristics (e. g. Technical alarm detection such as leads off or battery low and/or damaged, Watchdog timer implementation etc.)
- Safety requirements (e. g. Password protection, Function prioritization, Default settings, Verification and secure storage of patient information and physiological parameter information, network security, software risk controls such as software runaway etc.)
- timing and memory requirements
- Possible and/or probable changes in functionality (e.g. Display window resizing while viewing trend data, change User interface or screen layout when operating mode is changed e.g. in ventilator, pressure mode / volumetric mode etc.)
- Interoperability requirements.

The software component description shall take into consideration:

- **Allocation of RISK CONTROL measures for software.** (For e. g.: For class C software, it is mandatory to identify the separation between software items / components that are essential to risk control etc.)
- Failure modes of components and their effects; (e.g. Data corruption, device malfunctioning etc.)
- Common cause failures; (e.g. EMI / EMC, incorrect device handling, network failure, software virus, divide by zero, off by one, numeric overflow, incorrect memory management etc.)
- Systematic failures; test interval duration and diagnostic coverage; (e. g. Provide warning message / alarms, system go into safe mode when any failure is detected etc.)
- Maintainability; (e. g. provision for version upgrade, maintaining service and error log, modular coding, reuse of components, conformance to coding standards, providing adequate comments etc.)
- Protection from reasonably foreseeable misuse; (e. g. protection against incorrect settings, role based password protected access, provide confirmation messages, warning messages etc.)
- The NETWORK/DATA COUPLING specification, if applicable. >>

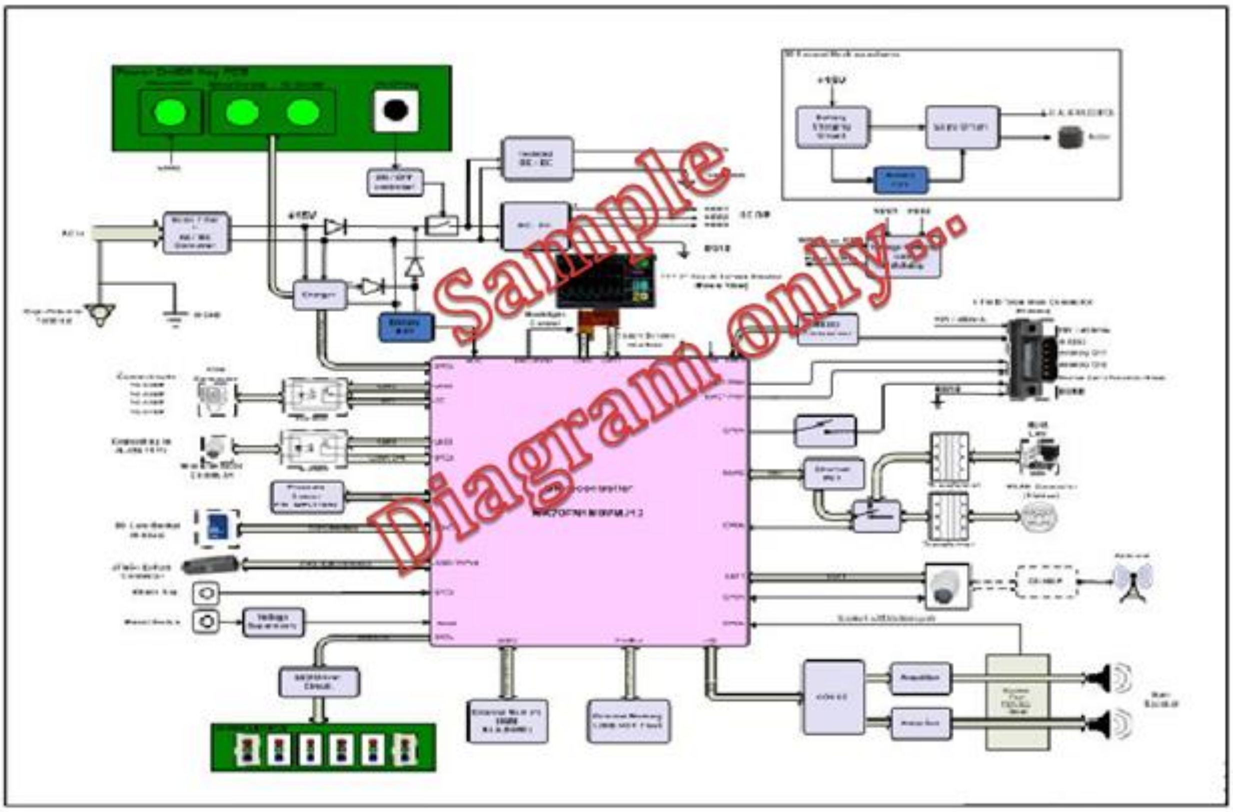
2.2.2. Software Component 2 description

<<Repeat the pattern for each top level component. The description of the components/ items can be very brief if it is present in the SDD.] (E.g.: In case of vital sign patient monitoring, ECG acquisition can be one component, in case of dialysis machine dialysate monitoring, in case of ventilator delivery of set tidal volume can be one component, in case of blood aphaeresis machine Extra Corporeal Blood Volume computation can be one component etc.>>

2.3. Physical architecture overview

<<This section should provide a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components. **Don't go into too much detail about the individual**

components themselves (there is a subsequent section for detailed component descriptions). The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality. In other words please provide description about each system block defined in section ‘System Overview’.



- Describe how the system is broken down into its components/subsystems (identifying each top-level component/subsystem/ components/units and the roles/responsibilities assigned to it).
- Describe how the higher-level components collaborate with each other in order to achieve the required results.
- Diagrams that describe a particular component or subsystem should be included within the particular subsection that describes that component or subsystem.
- Describe the hardware components on which software runs and their interactions/relationships. (Use components diagrams, deployment diagrams, network diagrams, interface diagrams if needed)
- Design Environment;(This should describe the development methods (this can be briefly described and later explained in the corresponding plans), CASE tools, hardware and designing platform, simulation and design tools, design development standards.)>>

2.3.1. Hardware Component 1 description

<<Provide **brief** description about this hardware component. While preparing this description, the following parameters are to be considered (but not limited to) based on the design and requirement at hand;

- Top level component identification (E.g. Battery management module, SMPS, DC-DC module/ Board, Processor board/ module, etc.)
- The purpose of the component. (E.g. in Battery management module:- to charge battery, control battery charging rate and mode, communicate status to main functional component etc.)
- Mention whether the hardware component is to be purchased, re-used or built.
- Explanation of hardware structure within that module / component, including its input and output interfaces to other internal & external modules. (E. g. In Battery management module, provide description about charger section, battery voltage monitoring section, temperature and charge level status monitoring section etc.)

- The hardware resources it uses, (E.g. in Battery management module, use of separate controller and its features). Other examples that can be considered are memory size, I/O lines, protective components etc…
- Reliability / redundancy; redundancy is the duplication of critical components / functions of a system with the intention of increasing reliability of the system, usually in the case of a backup or fail-safe for basic safety or performance of the device (E. g. 1:- In Battery management module, achieving this functionality using independent dedicated controller in sync with main controller. In case of this independent dedicated controller failure, main controller will take charge and raise an alarm, E. g. 2:- In case of ventilators, providing parallel output driver for main pump motor. Providing two pressure sensors in parallel to detect non-invasive blood pressure, E. g. 3:- Providing two air detectors in series for air detection in case of invasive blood pressure monitoring etc.)
- Defensive design; (E. g. limiting output power on potentially hazardous outputs by restricting the delivered output power by means of filters, clamping circuits, providing power limiting parts, introducing means to limit the travel of actuators etc.)
- Internal hardware tests, calibration and checks (E.g. providing reference calibration signals, baseline setting etc.)
- Device limitations due to hardware (E.g. Execution speed, response time, baud rate, power consumption, capacity limitation etc.)
- Fault detection, tolerance, and recovery characteristics (E.g. Technical alarm detection such as leads off or battery low and/or damaged, Watchdog timer implementation etc.)
- Safety requirements (E.g. fuse protection, protection against surge voltages, isolation, current limiting devices etc.)
- Interoperability requirements (E.g. support to different SpO2 sensor modules, support to different printers etc.)
- Components with high integrity characteristics; (Components assure to be fault free in relation to the safety requirements during normal use and reasonably foreseeable misuse.) **Do practice caution while declaring components as high integrity components**.)
- Fail-safe functions; Fail-safe or fail-secure describes a device or feature which, in the event of a failure, responds in a way that will cause no harm or a very minimum of harm to other devices or danger to personnel. (This minimal level of harm should be mitOrganisationd in the risk management process to see whether it come below the “level of acceptance” criteria.) (e. g. 1 Solenoid valve put in drug delivery track, shuts down in case of device power failure, e. g. 2 Controller output goes in inactive state in case controller hangs etc.)
- Partitioning of functionality; (Partitioning of functionality with respect to the hardware also needs to be evaluated and documented, so that the separation of duty among each section is clearly identified.)

The hardware component description shall also take into consideration:

- **Allocation of RISK CONTROL measures for hardware component.** (For e.g.; sensors, actuators, PESS and interfaces, etc.)
- Failure modes of components and their effects (e.g.: Unexpected loss of electrical / mechanical integrity, Deterioration in function (e.g. gradual occlusion of fluid / gas path, or change in resistance to flow, electrical conductivity) as a result of ageing, wear and repeated use Fatigue failure) ;
- Common cause failures (e.g.: Electrode cable unintentionally plugged into power line receptacle, poor isolation between live parts, exposure of dust or water particulates, voltage fluctuations, EMI / EMC etc.);
- Systematic failures;
- test interval duration and diagnostic coverage;
- Maintainability or serviceability (Access to parts / modules, maintaining unique wire harnesses to avoid connection mismatch, proper grounding);
- Protection from reasonably foreseeable misuse; (E. g. providing protection against twisting / turning by incorporating strain relief, designing to absorb shocks and vibration in case of free fall, providing means to move or carry the device such as providing handle or wheels, providing guards on surfaces to protect from infiltration etc.)
- The network/data coupling specification, if applicable.
- Device limitations due to software.>>

2.3.2. Hardware Component 2 description

<<Repeat the pattern for each top level component. >>

2.3.3. Hardware Component 3 description

<<Repeat the pattern for each top level component. >>

Company Confidential. © 2023 Capgemini. All rights reserved.

2.4. COTS

<< If you use either hardware or software COTS (Components off the Shelf), list them here.
For each COTS, describe:

- Its identification and version.
- Its purpose.
- Where it comes from: manufacturer, vendor, university, etc…
- Whether it is maintained by a third party or not.
- If this is executable,
  - what are the hardware / software resources it uses
  - Whether it is insulated in the architecture and why
- Its interfaces and data flows.
- If any risk controls were mitOrganisationd, it has to be made note of.>>

3. Functional behavior of architecture

<<State the functional requirements by the architecture for each main function of the system, add a description of the sequences / data flow that occur/ sequence diagrams/ collaboration diagrams (as required).
Functional behaviors include;

- **Functional Requirements**-The fundamental or essential subject matter of the product as defined by the customer. They describe what the product has to do or what processing actions it is to take.
- **Essential Performances Requirements**-Function (Performance related) that the equipment needs to meet as per design and regulatory requirements.
- **Aesthetic Requirements**-Aesthetic requirements are the non-functional properties that the features must have, such as 'look and feel requirements', operability, human factor and usability. These requirements are as important as the functional requirements for the product's success.
- **Dynamic Behavior-** (optional)>>

3.1 Workflow / Sequence 1

<<Repeat the pattern for each main function of the system.>>

3.2 Workflow / Sequence 2

<<Repeat the pattern for each main function of the system.>>

Company Confidential. © 2023 Capgemini. All rights reserved.

4. Justification of architecture

4.1 System architecture capabilities

<<Provide a brief description and rationale of the hardware/software architecture in terms of capabilities (as applicable) and acceptance criteria. Parameter can include but not limited to;

- Performances (for example response time, user mobility, data storage, or any functional performance which has an impact on architecture),
- User / patient safety,
- Protection against misuse,
- Maintenance (cold maintenance or hot maintenance),
- Adaptability, flexibility, upgradability, robustness,
- Scalability, availability,
- Backup and restore (both power and data),
- Hardware and Software security : fault tolerance, redundancy, emergency stop, recovery after crash …
- Administration,
- Monitoring,
- Miniaturization,
- Internationalization,
- Use of a particular type of product (programming language, database, library, etc. ...)
- Future plans for extending or enhancing the software/ hardware,
- User interface paradigms (or system input and output models)
- Hardware and/or software interface paradigms,
- Error detection and recovery,
- Memory management policies,
- External databases and/or data storage management and persistence,
- Distributed data or control over a network,
- Generalized approaches to control,
- Provisions for upgrading,
- Concurrency and synchronization,
- Communication mechanisms,
- Management of other resources,
- Interpretation,
- Compliance and regulatory. >>

4.2 Network architecture capabilities

<<If the medical device uses/has a network, describe here the rationale and acceptance criteria's of the hardware / network architecture:

- Bandwidth,
- Network failures,
- Loss of data,
- Inconsistent data,
- Inconsistent timing of data,
- Cyber security.>>

4.3 Risk analysis outputs

<<If the results of risk analysis have an impact on the architecture, describe here for each risk analysis output what has been done to mitOrganisation the risk in the architecture.
Use diagrams if necessary, e.g.: like architecture before risk mitigation and architecture after risk mitigation, to explain the choices. >>

4.4 Human factors & usability engineering outputs

<<If the results of human factors & usability analysis have an impact on the architecture, describe here for each risk human factors output .what has been done to meet Organisation the risk in the architecture. >>

Company Confidential. © 2023 Capgemini. All rights reserved.

Document information

Change history

Revision	Date	Changes	Author	Reviewer	Approver
1.0	dd.mm.YY	Text	Name, Role	Name, Role	Name, Role

Distribution list

Organization	Name and role	Action	Comments
Text	Name, Role	For action or For information	Text

File storage

Location	Administrator	Comments
Text	Name	Text

Company Confidential. © 2023 Capgemini. All rights reserved.

About Capgemini Engineering

Capgemini Engineering combines, under one brand, a unique set of strengths from across the Capgemini Group: the world leading engineering and R&D services of Altran – acquired by Capgemini in 2020 - and Capgemini's digital manufacturing expertise. With broad industry knowledge and cutting-edge technologies in digital and software, Capgemini Engineering supports the convergence of the physical and digital worlds. Combined with the capabilities of the rest of the Group, it helps clients to accelerate their journey towards Intelligent Industry. Capgemini Engineering has more than 52,000 engineer and scientist team members in over 30 countries across sectors including aeronautics, automotive, railways, communications, energy, life sciences, semiconductors, software & internet, space & defense, and consumer products.

Capgemini Engineering is an integral part of the Capgemini Group, a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of over 325,000 team members more than 50 countries. With its strong 55-year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2021 global revenues of €18 billion.

Get the Future You Want | www.capgemini.com/engineering

This document contains information that may be privileged
or confidential and is the property of the Capgemini Group.