



**VOLTA**<sup>®</sup>  
S.p.A.

Competenza  
ed entusiasmo per  
soluzioni tecniche

# Riferimenti normativi

Software per  
laboratorio

---

Alessandro Corniani  
[aco@volta.it](mailto:aco@volta.it)

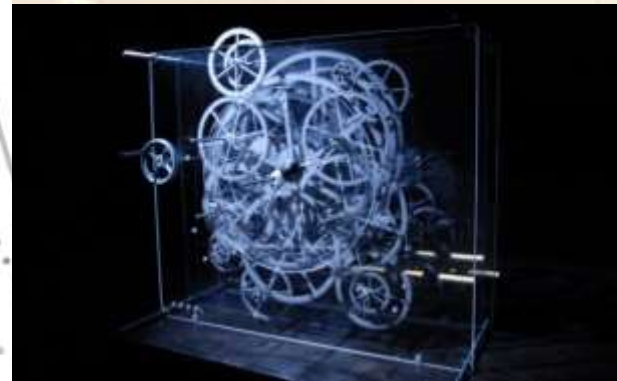


# Archimede

- Già nel De Re Publica (primo secolo BC), Cicerone scrive di sfere simulanti un planetario, progettate da Archimede e costruite da Talete di Mileto e Cnidio Eudoxo
- *Hanc sphaeram Galus cum moveret, fiebat ut soli luna totidem conversionibus in aere illo, quot diebus in ipso caelo succederet, ex quo et in caelo et in sphaera solis fieret eadem illa defectio*
- Nelle Synagoge, Pappo di Alessandria (quarto secolo AD), cita un libro di Archimede sulla produzione delle "sfere" che sarebbe stato l'unico libro su argomenti pratici scritto da Archimede

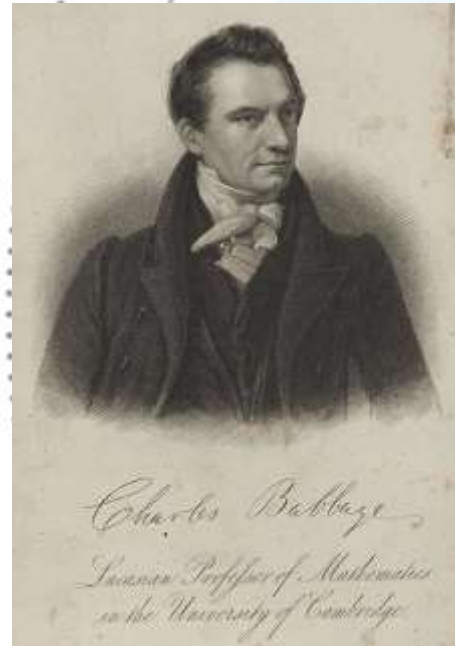
# Antikythera Mechanism

- Il meccanismo di Antikythera è il primo computer analogico arrivato a noi ed è databile al terzo secolo BC
- Simulava il movimento del sole e della luna, prediceva le eclissi e forniva le date dei giochi atletici panellenici



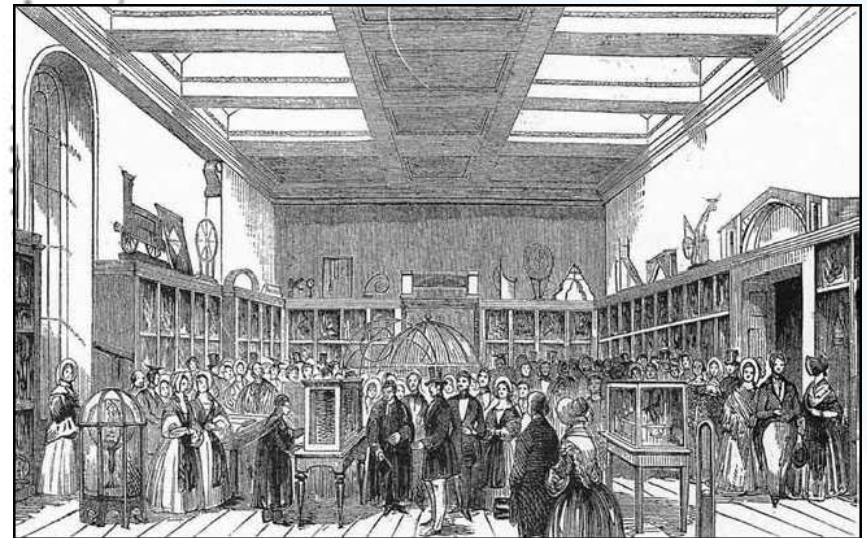
# Charles Babbage

- Nel giugno 1833, quando Charles Babbage incontra la diciassettenne Ada Byron, unica figlia legittima di Lord Byron



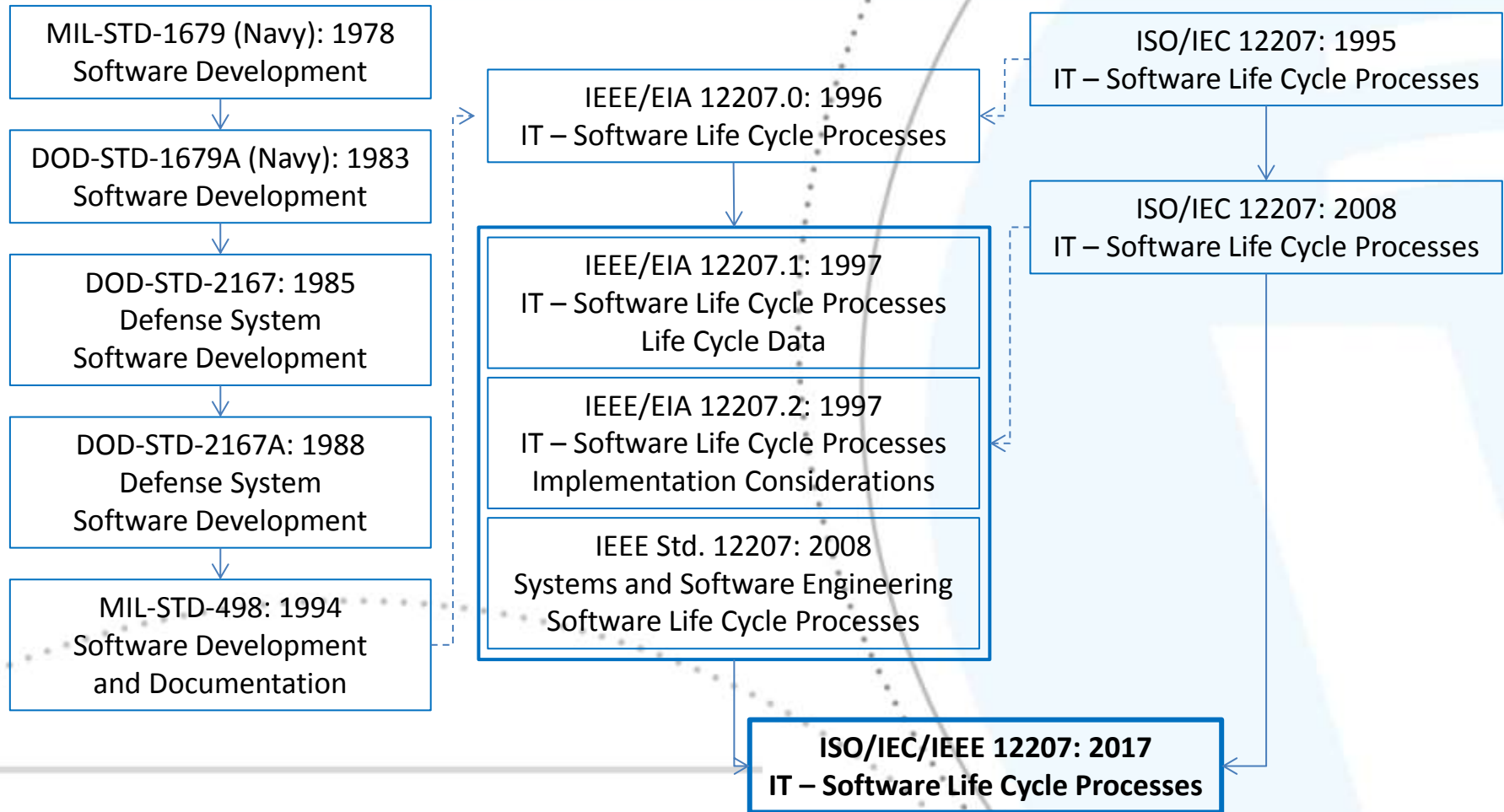
# Analytical Engine

- Charles Babbage esce da Cambridge nel 1814 come «top mathematician» ma senza onori e senza neppure discutere la tesi considerata blasfema
- Appassionato di astronomia, metrologia, crittografia, fu tra i fondatori della Royal Astronomy Society
- Pioniere del computer, sviluppa il primo «difference engine» e lo migliora come «analytical engine»

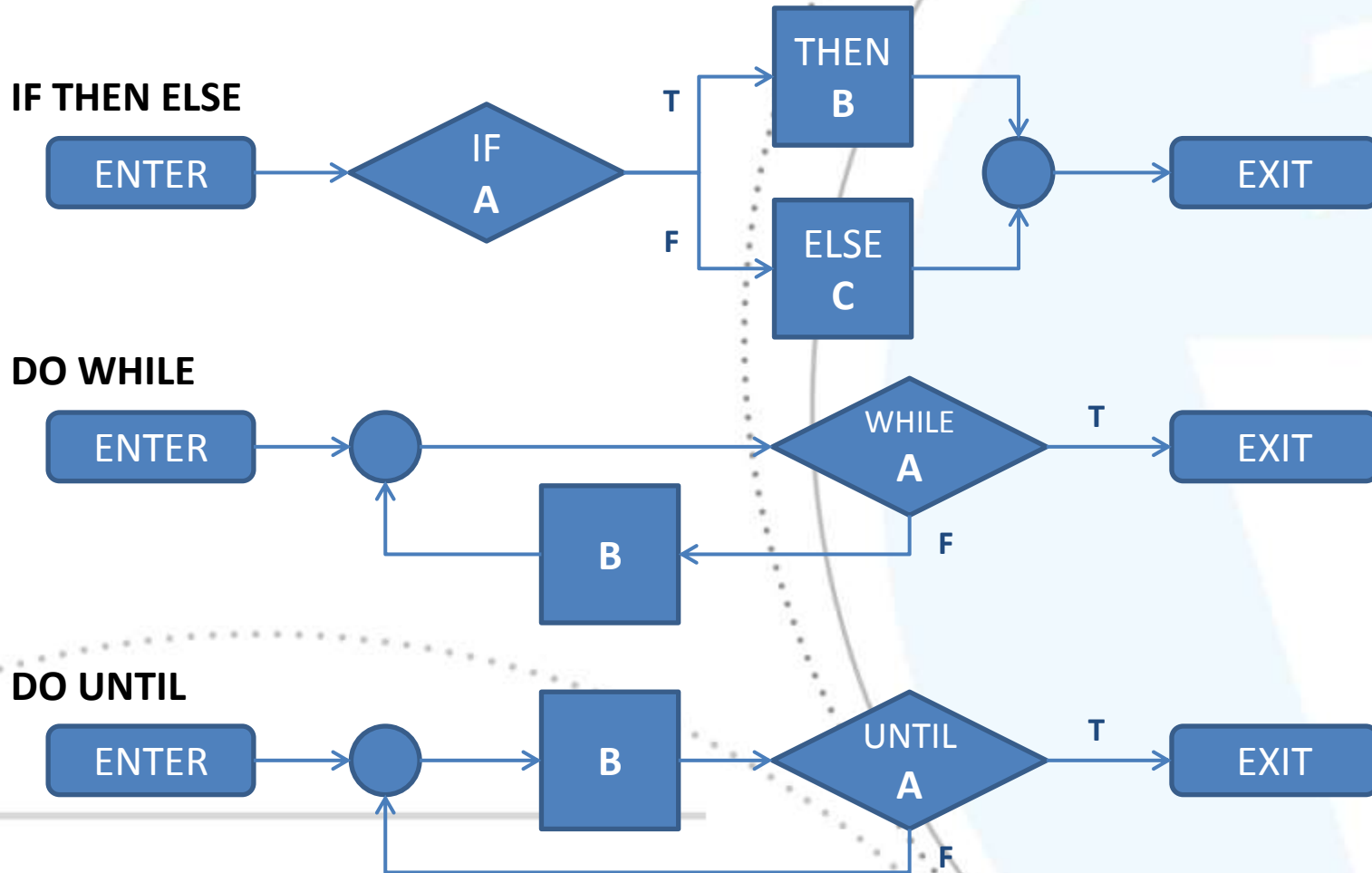




# Evoluzione delle Norme



# DOD-STD-1679A



# DOD-STD-1679A

- 5.8 – Software testing
  - 5.8.1 – Software integration and developmental testing pre-requisites
  - 5.8.2 – Software integration and developmental testing
  - 5.8.3 – System(s) integration test
  - 5.8.4 – Software trouble reporting
    - 5.8.4.1 – Software trouble report category
      - a. Software trouble (S)
      - b. Documentation trouble (D)
      - c. Design trouble (E)

# Qualità: Jim McCall

- Nel 1977, Jim McCall produce il primo modello di qualità del software per US Air Force.
- Per riuscire a caratterizzare costi e benefici di un software, McCall definisce un concetto rivoluzionario: un software di successo sarà quello che soddisfa i requisiti dell'utente in considerazione del punto di vista dello sviluppatore.
- Basi comuni:
  - costi e benefici sono visti nella loro totalità
  - la presenza o assenza di attributi del software è misurata in maniera oggettiva
  - il livello degli attributi del software fornisce un indice della qualità del software stesso
  - gli attributi del software ne permettono il miglioramento continuo

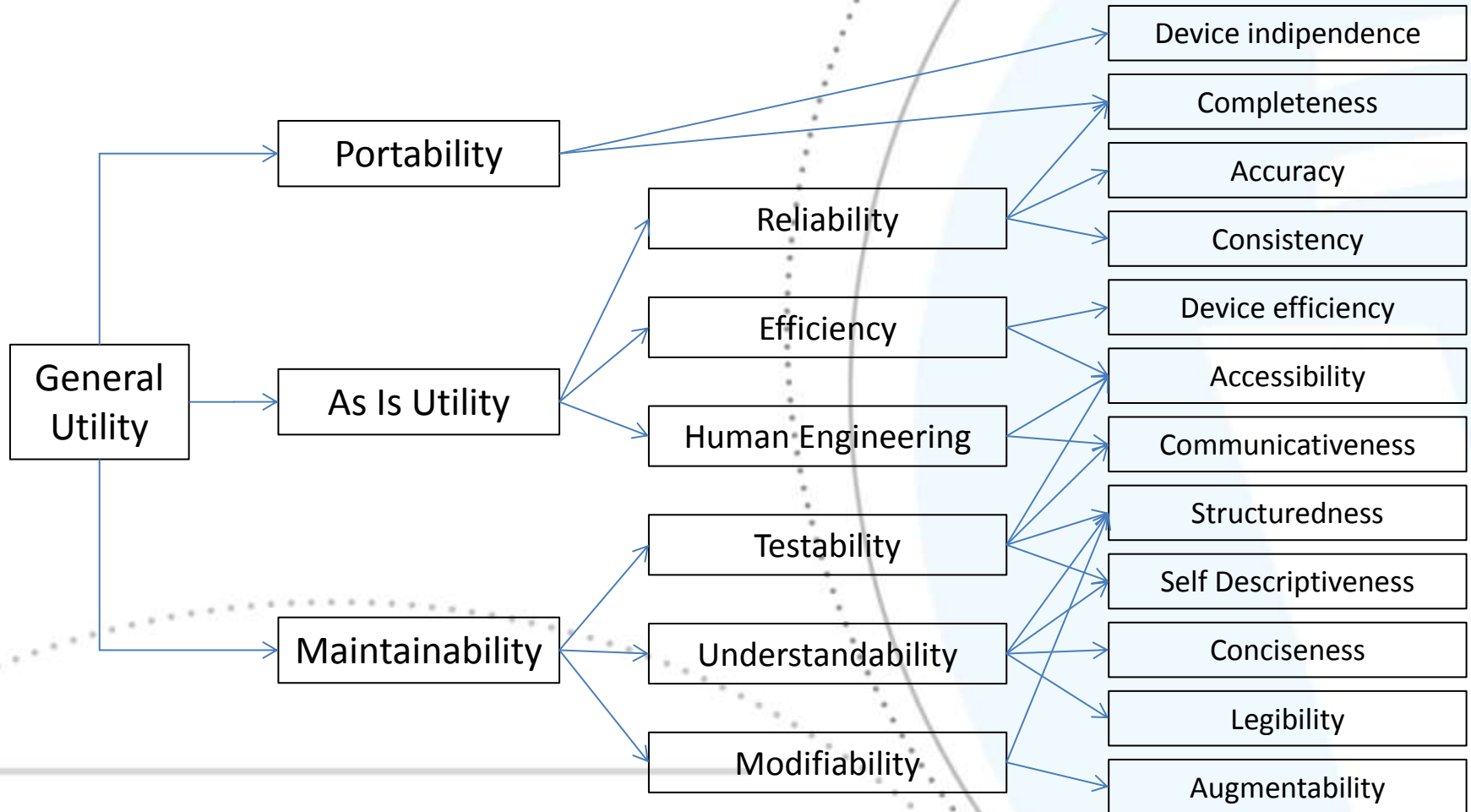
# Qualità: Jim McCall

- Perspectives:
  - Revision: abilità di modificare il software a seconda delle necessità future
    - Maintainability: trovare i difetti e correggerli
    - Flexibility: cambiare le specifiche nel tempo
    - Testability: validare i requisiti
  - Transition: adattabilità agli ambienti
    - Portability: trasferibilità tra ambienti diversi
    - Re-usability: riutilizzo di componenti software
    - Interoperability: comfort zone
  - Operations: funzionamento secondo specifiche
    - Correctness: il software segue le specifiche date
    - Reliability: la fiducia nel comportamento
    - Efficiency: l'utilizzo adeguato delle risorse
    - Usability: la facilità di utilizzo

# Qualità: Barry Boehm

- Già nel 1973, Barry Boehm predice a DARPA (Defense Advanced Research Project Agency) che i costi del software sarebbero andati ad essere molto maggiori di quelli dell'hardware, ispirando quindi l'inversione di paradigma e di investimenti in hardware
- Nel 1978, Boehm modifica il modello di McCall ed introduce la gerarchia delle caratteristiche del software: high level, intermediate e primitive.
- Per Boehm, le caratteristiche high level sono:
  - As-is utility: qual è lo sforzo che l'utente spende per imparare ad usare il software?
  - Maintainability: quanto è facile mantenere il software e modificarlo perché soddisfi futuri requisiti?
  - Portability: posso usarlo in diversi ambienti?

# Qualità: Barry Boehm



# IEEE Software life cycle

- IEEE 730-2014 – Software quality assurance processes–
- IEEE 828-2012 – Configuration management in systems and software engineering
- ISO/IEC/IEEE 29119 – Systems and software engineering – Software testing
  - ISO/IEC/IEEE 29119-1-2013 – Concepts and definitions
  - ISO/IEC/IEEE 29119-2-2013 – Test processes
  - ISO/IEC/IEEE 29119-3-2013 – Test documentation
  - ISO/IEC/IEEE 29119-4-2013 – Test techniques
  - ISO/IEC/IEEE 29119-5-2013 – Keyword-driven testing
- IEEE 830-1998 – Recommended practice for software requirements specification
- IEEE 1012-2016 – System, software and hardware verification and validation
- IEEE 1016-2009– IT – Systems design - Software design descriptions
- ISO/IEC/IEEE 16326-2009 – Systems and software engineering – Life cycle processes – Project management
- IEEE 26511-2012 – Systems and software engineering – Requirements for managers of user documentation
- IEEE 26512-2011 – Systems and software engineering – Requirements for acquirers and suppliers of user documentation

# Validation

- Barry Boehm:
  - Validation: are we building the product right?
- IEEE-STD-610
  - Validation: the process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.
- ISO/IEC 12207:2008
  - 4.54 – Validation: confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ISO 9000:2005]
    - Note: validation in a life cycle context is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives.
- ISO 9000:2015
  - 3.8.13 – Validation: confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled
    - Note 1: The objective evidence needed for a validation is the result of a test or other form of determination such as performing alternative calculations or reviewing documents.
    - Note 2: The word “validated” is used to designate the corresponding status.
    - Note 3: The use conditions for validation can be real or simulated.
- **La validazione durante il processo di sviluppo del software assicura che l’output di ogni fase del software effettivamente sviluppi correttamente quanto richiesto e che il prodotto sia stato progettato correttamente.**

# Verification

- Barry Boehm:
  - Verification: are we building the right product?
- IEEE-STD-610
  - Verification: the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
- ISO/IEC 12207:2008
  - 4.55 – Verification: confirmation, through the provision of objective evidence, that specified requirements have been fulfilled [ISO 9000:2005]
    - Note: Verification in a life cycle context is a set of activities that compares a product of the life cycle against the required characteristics for that product. This may include, but is not limited to, specified requirements, design description and the system itself.
- ISO 9000:2015
  - 3.8.12 – Verification: confirmation, through the provision of objective evidence, that specified requirements have been fulfilled
    - Note 1: The objective evidence needed for a verification can be the result of an inspection or of other forms of determination such as performing alternative calculations or reviewing documents.
    - Note 2: The activities carried out for verification are sometimes called a qualification process.
    - Note 3: The word “verified” is used to designate the corresponding status.
- **La verifica si può considerare una ricerca di codice mancante o errato, di errori durante l’esecuzione e di mancanza di corrispondenza ai requisiti di base.**

# Accredia DT-02-DT

- Guida alla gestione e controllo del sistema informativo dei laboratori
  - Scelta del software ed organizzazione del sistema informativo
  - Installazione del software
  - Verifica funzionale dell'hardware e del software
  - Corretto impiego e mantenimento dello stato di servizio del software
  - Sicurezza dei dati (controlli sulla correttezza, salvataggio e distribuzione dei risultati)

# Accredia DT-02-DT

- Software commerciale
  - Si tratta di software acquistato «as it is» da un rivenditore o da un produttore, senza che l'utente possa modificarne le funzioni previste dal progettista. L'esigenza del laboratorio si adatta in modo soddisfacente alle caratteristiche funzionali garantite.
  - In questa categoria rientrano i sistemi operativi, i programmi di gestione di testo, di calcoli e di gestione dati, i browser, i client di posta elettronica, i firewall, gli antivirus, i programmi di sviluppo delle applicazioni di automazione e i programmi forniti dai produttori di strumenti a corredo per il controllo tramite PC.

# Accredia DT-02-DT

- Software sviluppato
  - Si tratta di software sviluppato dal personale del laboratorio o su specifiche fornite dal laboratorio stesso da parte di terzi per realizzare una funzione non altrimenti raggiungibile. L'esigenza del laboratorio è tale da richiedere lo sviluppo di uno o più componenti che si devono adattare adeguatamente ai requisiti richiesti.
  - In questa categoria rientrano i fogli di calcolo sviluppati con programmi di gestione di calcoli (e.g. cartelle di Microsoft Excel o progetti di MatLab), database personalizzati e programmi di automazione sviluppati con linguaggi di programmazione (e.g. Microsoft Visual Basic, C, Java, Cec Testpoint, National LabView...)

# Requisiti di ISO 17025

- La norma ISO 17025 introduce in modo esplicito l'utilizzo del calcolatore nel laboratorio per:
  - gestione della documentazione
  - gestione ed esecuzione delle misure e delle prove
  - redazione e gestione dei certificati e dei rapporti di misura e di prova
  - raccolta ed archiviazione dei dati di misura, dei risultati e delle apparecchiature impiegate nel laboratorio
  - trasmissione dei risultati

# Processi critici

- Archiviazione delle registrazioni
  - Apparecchiature di misurazione ed attrezzature
  - Strumenti
  - Documenti
- Processi da validare
  - Controllo accessi
  - Sicurezza dei dati

# Processi critici

- Generazione del risultato
  - Impostazione delle misure
  - Acquisizione dati
  - Controllo delle misure
  - Calcolo
- Processi da validare
  - Software di comunicazione tra strumenti
  - Software di gestione della misura
  - Software di calcolo
  - Funzioni di archiviazione dei parametri utilizzati

# Schede tecniche software

- Software sviluppato
  - Identificativo: numero di serie, numero e data di revisione, licenza...
  - Tipo di software
  - Unità operativa di appartenenza
  - Descrizione funzionale e dei manuali
  - Descrizione della procedura di validazione e di collaudo e di conferma (verifiche funzionali)
  - Dati riguardanti le attività di validazione, di collaudo e di conferma (data, operatore, esito...)

# Schede tecniche software

- Software commerciale
  - Identificativo: numero di serie, numero e data di revisione, licenza...
  - Tipo di software
  - Valore dei principali parametri di configurazione
  - Unità operativa di appartenenza
  - Descrizione della procedura di collaudo
  - Dati riguardanti le attività di collaudo (data, operatore, esito...)
  - Politica di aggiornamento

# Validazione e collaudo

- Software sviluppato:
  - verifica sui formati: si presenta come atteso?
  - verifica sui componenti: ci sono tutti?
  - collaudo
- Dopo la validazione ed il collaudo, occorre registrare l'esito nelle schede del software

# Validazione e collaudo

- Software commerciale:
  - collaudo
- A seguito di collaudo positivo, il software può essere utilizzato

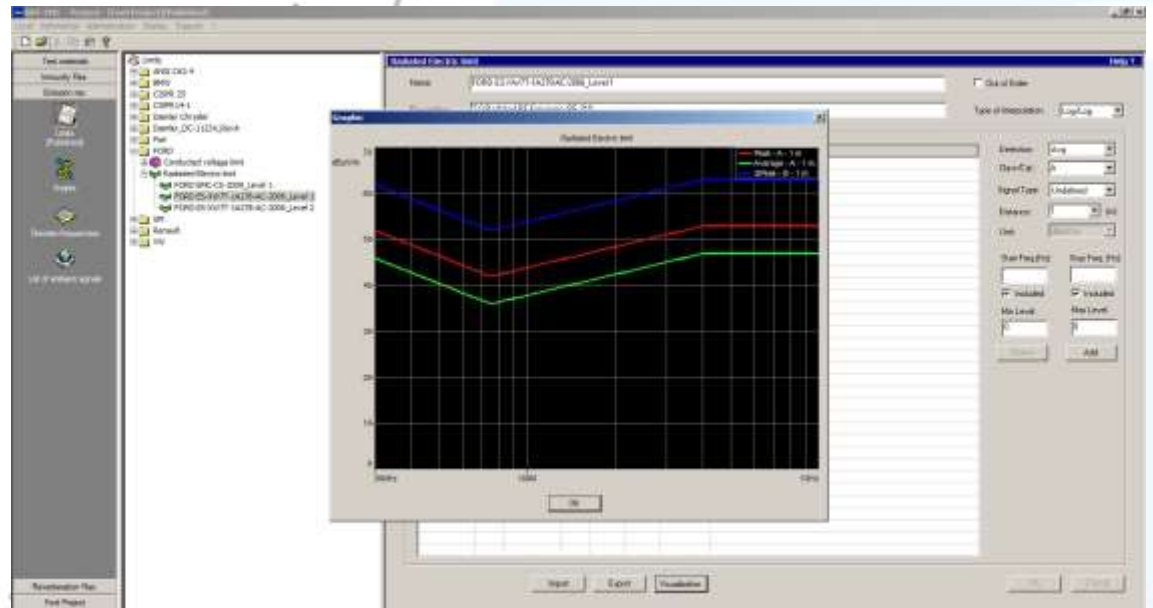
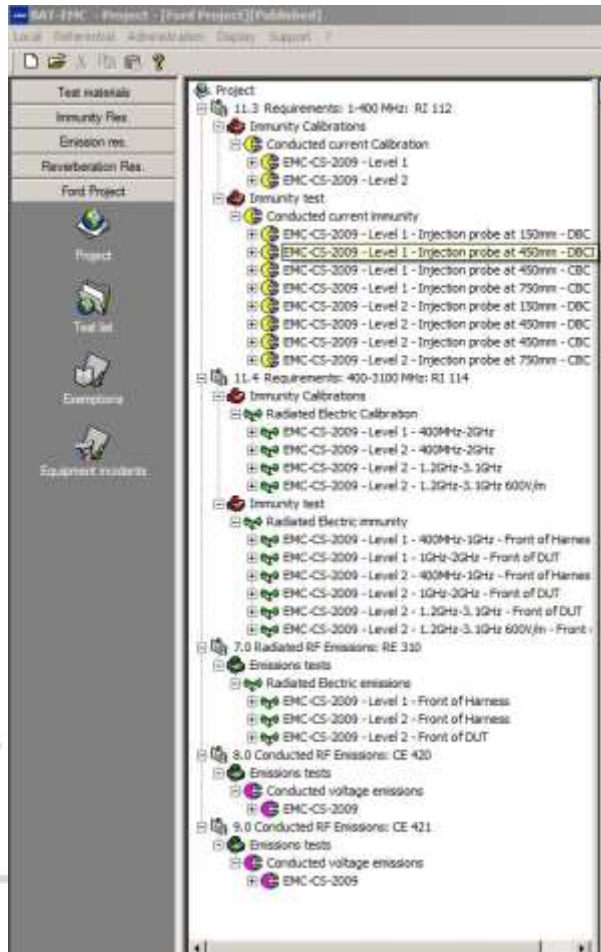
# Eurolab guidance

- Tre categorie:
  - COTS: Commercial Off The Shelf
    - OS: Windows, Linux
    - Firmware: strumentazione
    - Pacchetti: Word, Excel, Outlook, software di controllo della strumentazione fornito
  - MOTS: Modified Off The Shelf
    - Programmi su misura a partire da ambienti COTS: formule Excel, programmi LabView...
  - CUSTOM
    - Programmi scritti in linguaggi di programmazione: C++, Java, Visual Basic, XML, Word ed Excel VBA code...

# Eurolab guidance

- **COTS:**
  - Verifica che il software acquistato sia conforme ai requisiti
  - Collaudo
- **MOTS e CUSTOM:**
  - Verifica puntuale dei requisiti con prove di funzionalità su processi critici
  - Collaudo

# COTS: BAT-EMC



# COTS: autowave.control

The screenshot displays the autowave.control software interface. At the top, there is a menu bar with 'Setup', 'View', 'Service', and 'Help'. A red status indicator shows '< Simulation >'. Below the menu, there are several functional buttons: 'Play' (green), 'Record' (blue), 'Play & Record' (red), 'User Tests' (orange), and 'Reports' (grey). The main workspace is divided into 'Single Channel Play' and 'Multi Channel Play' sections. A 'Standard Test Select' dialog box is open, showing a list of manufacturers under the 'Name' column. The list includes Audi, BMW, Case New Holland, Caterpillar, China Motor Company, Chrysler, Claas, CTCT, Cummins, DAF, Daihatsu, DaimlerChrysler, Ducati, FCA (Fiat Chrysler Automotive), FAW (First Automotive Works), Fiat, Ford, Freightliner, General Motors, and Harley-Davidson Motor Company. The dialog box has 'List View', 'Tree View', and 'Search Table' tabs, and 'Select' and 'Cancel' buttons at the bottom.

autowave.control

Setup View Service Help

< Simulation >

AutoWave

GENERAL

Play Record Play & Record User Tests Reports

Single Channel Play | Multi Channel Play

Standard Test Select

List View Tree View Search Table

Name

- Manufacturer
  - Audi
  - BMW
  - Case New Holland
  - Caterpillar
  - China Motor Company
  - Chrysler
  - Claas
  - CTCT
  - Cummins
  - DAF
  - Daihatsu
  - DaimlerChrysler
  - Ducati
  - FCA (Fiat Chrysler Automotive)
  - FAW (First Automotive Works)
  - Fiat
  - Ford
  - Freightliner
  - General Motors
  - Harley-Davidson Motor Company

Select Cancel

# Accredia DT-02-DT

- **Mantenimento dello stato di servizio**
  - Aggiornamenti previsti
  - Documentazione
- **Dismissione**
  - Messa fuori servizio della copia di installazione del software
  - Disinstallazione
  - Documentazione

# ISO 17025:2017

- Quali punti possono essere importanti nella norma ISO 17025 quando si parla di software oltre a quelli già analizzati da DT-02-DT?

# ISO 17025:2017

- Definisce i requisiti per i laboratori di prove e di misure
- Armonizza le procedure mettendo le basi per facilitare la co-operazione tra laboratori ed altri enti ed assistere lo scambio di informazioni e di esperienze

# ISO 17025:2017

- 3.6 – La definizione di laboratorio comprende le attività di prova, di misura e di campionamento

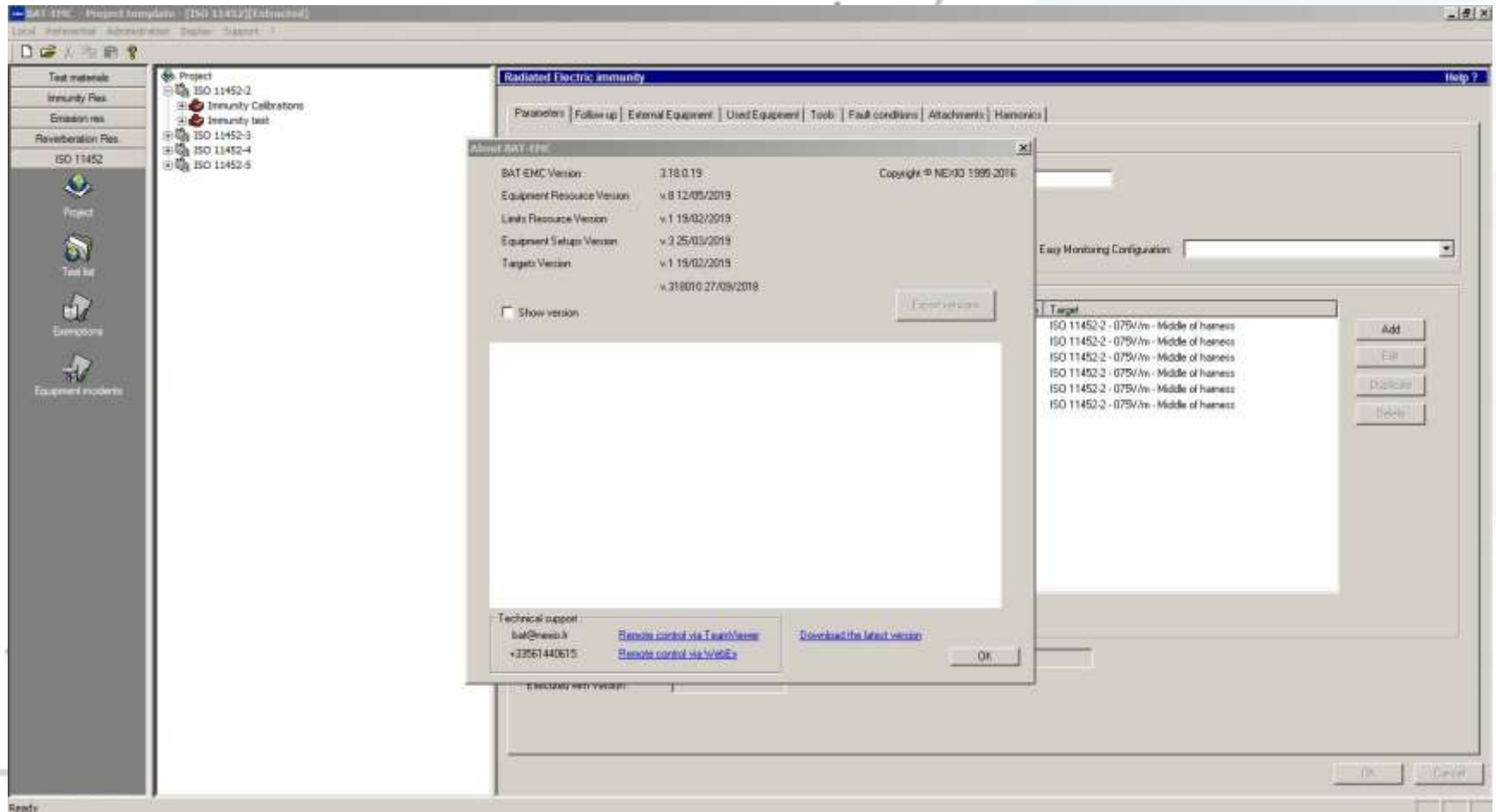
# ISO 17025:2017

- 6.4.13 – Occorre registrare tutti i parametri degli strumenti che influenzano le attività di laboratorio
  - Identificare la strumentazione
  - Dare evidenza della verifica
  - Tenere traccia delle date e degli esiti delle tarature
  - Tenere traccia di danni, malfunzionamenti o riparazioni

# ISO 17025:2017

- 7.5 – La documentazione deve contenere tutte le informazioni che possano facilitare l'identificazione di fattori che influenzano la prova e l'incertezza di misura e permettano la ripetibilità della prova; i dati devono essere registrati e facilmente identificabili successivamente

# ISO 17025:2017



# ISO 17025:2017

- 8 – Sistema di gestione
  - Il laboratorio deve implementare il controllo necessario all'identificazione, memorizzazione, protezione, back-up, archivio, recupero, tempo di ritenzione e smaltimento della documentazione



**VOLTA**<sup>®</sup>

S.p.A.

Competenza  
ed entusiasmo per  
soluzioni tecniche

