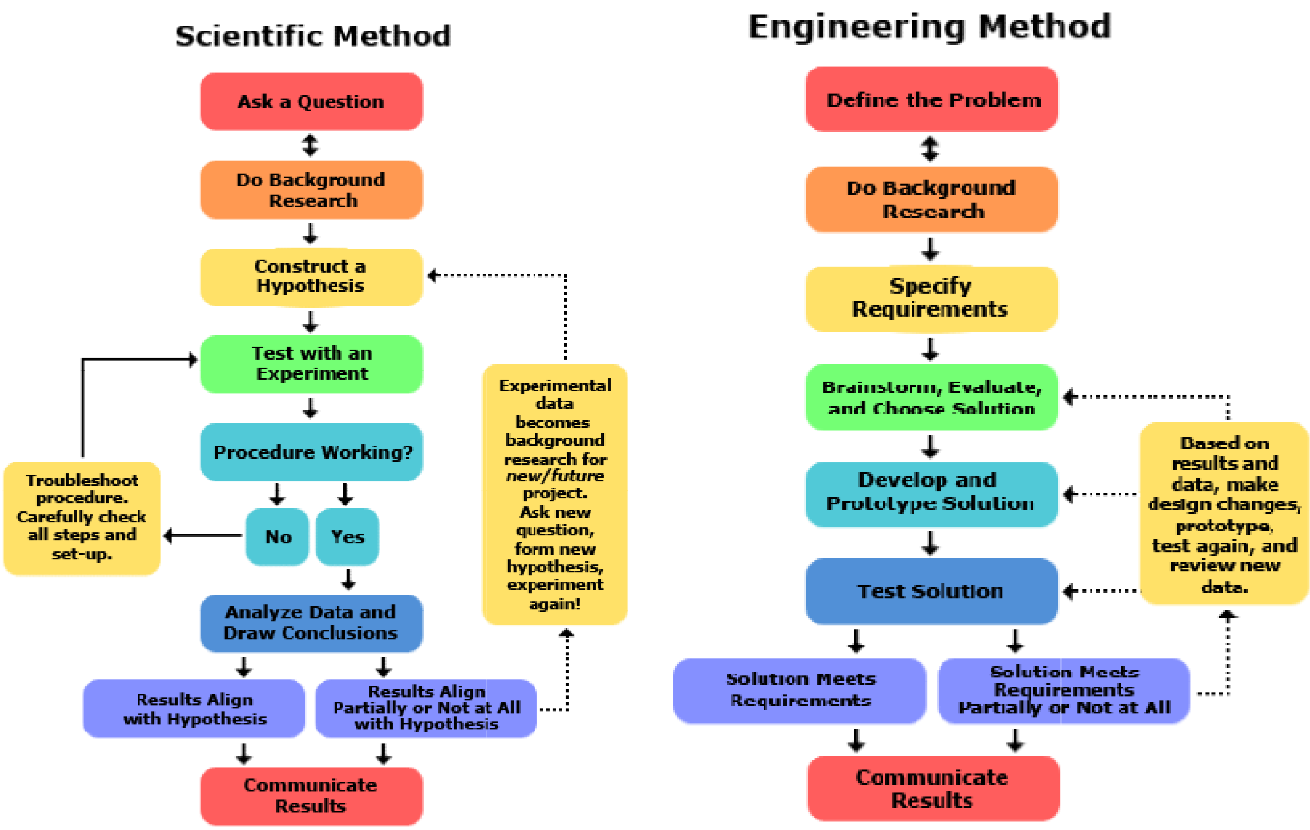
UNIT IV

METHODOLOGY OF ENGINEERING

DIFFERENCE BETWEEN SCIENTIFIC METHOD AND ENGINEERING DESIGN

While scientists study how nature works and discover new knowledge about the universe, engineers create or construct new things, such as products, websites, environments, and experiences. Because engineers and scientists have different objectives, they follow different processes in their work. Scientists perform experiments using the scientific method; whereas engineers follow the creativity-based engineering design process. You can see the steps of each process in these flowcharts:



The Scientific Method	The Engineering Design Process
State your question	Define the problem
Do background research	Do background research
Formulate your hypothesis, identify variables	Specify requirements
Design experiment, establish procedure	Create alternative solutions, choose the best one and develop it

The Scientific Method	The Engineering Design Process
Test your hypothesis by doing an experiment	Build a prototype
Analyze your results and draw conclusions	Test and redesign as necessary
Communicate results	Communicate results

Why are there two processes?

Both scientists and engineers contribute to the world of human knowledge, but in different ways. Scientists use the scientific method to make testable explanations and predictions about the world. A scientist asks a question and develops an experiment, or set of experiments, to answer that question. Engineers use the engineering design process to create solutions to problems. An engineer identifies a specific need: **Who** need(s) **what** because **why**? And then, he or she creates a solution that meets the need.

Which process should I follow for my project?

Watch the video to see what it looks like to tackle the same topic using the scientific method versus the engineering design process.

In real life, the distinction between science and engineering is not always clear. Scientists often do some engineering work, and engineers frequently apply scientific principles, including the scientific method. Much of what we often call "computer science" is actually engineering—programmers creating new products. Your project may fall in the gray area between science and engineering, and that's OK. Many projects, even if related to engineering, can and should use the scientific method.

However, if the objective of your project is to invent a new product, computer program, experience, or environment, then it makes sense to follow the engineering design process.

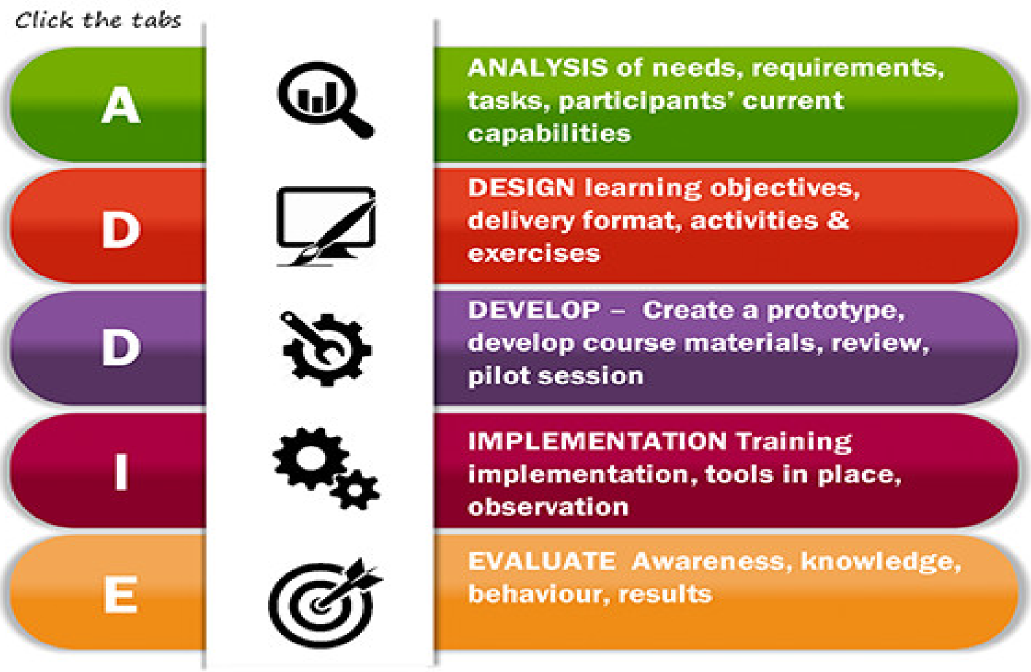
ADDIE MODEL:

The ADDIE model is the generic process traditionally used by instructional designers and training developers. The five phases—Analysis, Design, Development, Implementation, and Evaluation—represent a dynamic, flexible guideline for building effective training and performance support tools. While perhaps the most common design model, there are a number of weaknesses to the ADDIE model which have led to a number of spin-offs or variations.
It is an Instructional Systems Design (ISD) model. Most of the current instructional design models are spin-offs or variations of the ADDIE model; other models include the Dick & Carey and Kemp ISD models. One commonly accepted improvement to this model is the use of rapid prototyping. This is the idea of receiving continual or formative feedback while instructional materials are being created. This model attempts to save time and money by catching problems while they are still easy to fix.

Instructional theories also play an important role in the design of instructional materials. Theories such as behaviorism, constructivism, social learning and cognitivism help shape and define the outcome of instructional materials.

In the ADDIE model, each step has an outcome that feeds into the subsequent step.

Analysis > Design > Development > Implementation > Evaluation



Analysis Phase

In the analysis phase, instructional problem is clarified, the instructional goals and objectives are established and the learning environment and learner's existing knowledge and skills are identified. Below are some of the questions that are addressed during the analysis phase:

* Who is the audience and their characteristics?
* Identify the new behavioral outcome?
* What types of learning constraints exist?
* What are the delivery options?
* What are the online pedagogical considerations?
* What is the timeline for project completion?

Design Phase

The design phase deals with learning objectives, assessment instruments, exercises, content, subject matter analysis, lesson planning and media selection. The design phase should be systematic and specific. Systematic means a logical, orderly method of identifying, developing and evaluating a set of planned strategies targeted for attaining the project's goals. Specific means each element of the instructional design plan needs to be executed with attention to details.

These are steps used for the design phase:

* Documentation of the project's instructional, visual and technical design strategy
* Apply instructional strategies according to the intended behavioral outcomes by domain

(cognitive, affective, psychomotor).
* Create storyboards
* Design the user interface and user experience
* Prototype creation
* Apply visual design (graphic design)

Development Phase

The development phase is where the developers create and assemble the content assets that were created in the design phase. Programmers work to develop and/or integrate technologies. Testers perform debugging procedures. The project is reviewed and revised according to any feedback given.

Implementation Phase

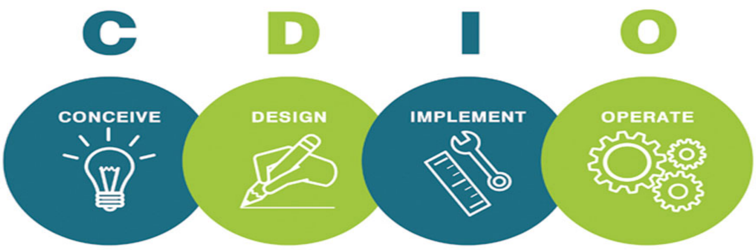
During the implementation phase, a procedure for training the facilitators and the learners is developed. The facilitators' training should cover the course curriculum, learning outcomes, method of delivery, and testing procedures. Preparation of the learners include training them on new tools (software or hardware), student registration.

This is also the phase where the project manager ensures that the books, hands on equipment, tools, CD-ROMs and software are in place, and that the learning application or Web site is functional.

Evaluation Phase

The evaluation phase consists of two parts: formative and summative. Formative evaluation is present in each stage of the ADDIE process. Summative evaluation consists of tests designed for domain specific criterion-related referenced items and providing opportunities for feedback from the users.

CDIO ENGINEERS IN INDUSTRY



Conceive:

- Defining Customer needs
- Considering technology
- Enterprise Strategy and regulations
- Developing Concepts, techniques and
- Business Plan

Design:

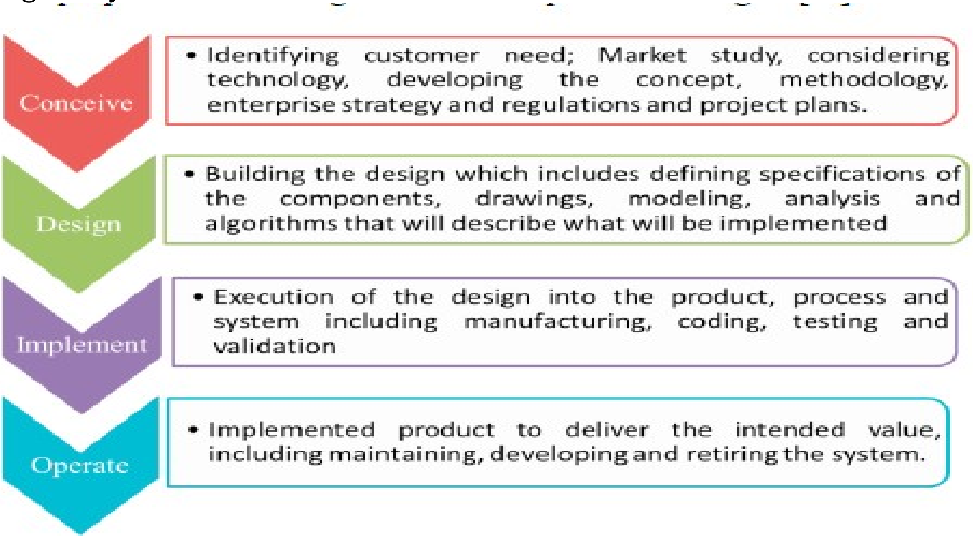
- Creating the design
- The plans, drawings and algorithms that describe what will be implemented

Implement:

- The transformation of design into the product, including manufacturing, coding , testing and validation

Operate:

- Using the implemented product to deliver the intended values, including maintaining, evolving and retiring the system



ENGINEERING DESIGN PROCESS

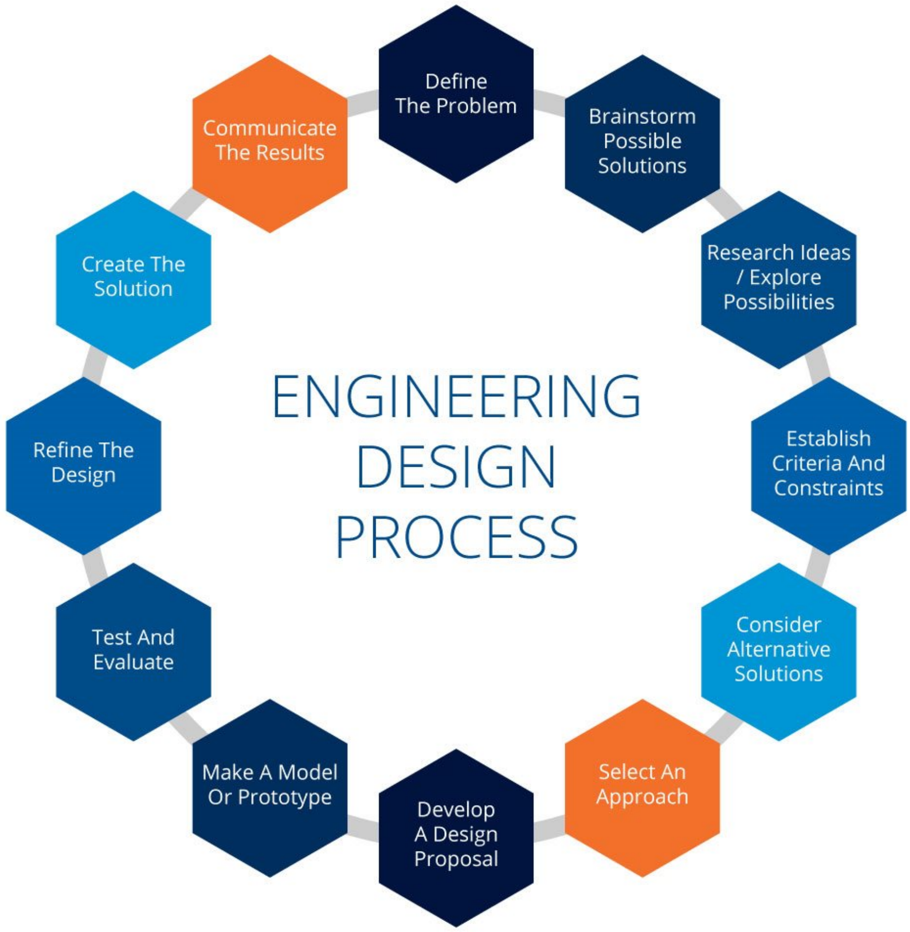
The engineering design process is a series of steps that engineers follow to find a solution to a problem. The steps include problem solving processes such as, for example, determining your objectives and constraints, prototyping, testing and evaluation.

The process is important to the work conducted by TWI and is something that we can offer assistance with.

While the design process is iterative it follows a predetermined set of steps, some of these may need to be repeated before moving to the next one. This will vary depending on the project itself, but allows lessons to be learnt from failures and improvements to be made.

The process allows for applied science, mathematics and engineering sciences to be used to achieve a high level of optimisation to meet the requirements of an objective. The steps include problem solving processes such as, for example, determining your objectives and constraints, prototyping, testing and evaluation.

The steps of the engineering process are not always followed in sequence, but it is common for engineers to define the problem and brainstorm ideas before creating a prototype test that is then modified and improved until the solution meets the needs of the engineers project. This is called iteration and is a common method of working.



1. Define The Problem

What is the problem that needs to be solved? Who is the design product for, and why is it important to find a solution? What are the limitations and requirements? Engineers need to ask these types of critical questions regardless of what is being created.

2. Brainstorm Possible Solutions

Good designers brainstorm possible solutions before opting to start a design, building a list of as many solutions as possible. It is best to avoid judging the designs and instead just let the ideas flow.

3. Research Ideas / Explore Possibilities for your Engineering Design Project

Use the experience of others to explore possibilities. By researching past projects you can avoid the problems faced by others. You should speak to people from various backgrounds, including users or customers. You may find some solutions that you had not considered.

4. Establish Criteria and Constraints

Having listed potential solutions and determined the needs of the project alongside your research, the next step is to establish any factors that may constrain your work. This can be done by revisiting the requirements and bringing together your findings and ideas from previous steps.

5. Consider Alternative Solutions

You may wish to consider further solutions to compare the potential outcomes and find the best approach. This will involve repeating some of the earlier steps for each viable idea.

6. Select An Approach

Once you have assessed your various options you can determine which approach best meets your requirements. Reject those that don't meet your requirements.

7. Develop A Design Proposal

Having chosen your approach, the next step is to refine and improve the solution to create a design proposal. This stage can be ongoing through the length of your project and even after a product has been delivered to customers.

8. Make A Model Or Prototype

Use your design proposal to make a prototype that will allow you to test how the final product will perform. Prototypes are often made from different materials than the final version and are generally finished to a lesser standard.

9. Test And Evaluate

Each prototype will need testing, re-evaluation and improvement. Testing and evaluation allows you to see where any improvements are needed.

10. Refine The Design

Once testing has been completed, the design can be revised and improved. This step can be repeated several times as more prototypes are created and evaluated.

11. Create The Solution

After your refinements have been completed and fully tested, you can decide upon and create your finished solution. This may take the form of a polished prototype to demonstrate to customers.

12. Communicate The Results

The final stage is to communicate your results. This can be in the form of a report, presentation, display board, or a combination of methods. Thorough documentation allows your finished product to be manufactured to the required quality standards.

OPERATIONAL FACTORS IN SYSTEM DESIGN

One of the most intriguing aspects of software architecture is trying to bring structure to areas that can't be structured easily. Whenever an architect designs a system, service, or feature, they are formulating a typical yet comprehensive solution to a unique problem.

The key concepts outlined here are valuable in designing an efficient, scalable, accessible, secure, and cost-friendly architecture.

Integrity and Consistency

The integrity of the data the system operates on is of the highest consideration when designing a reliable and fault-tolerant architecture. The system should be designed to provide redundant backups that maintain data integrity and all-around consistency.

Performance and Scalability

Modern web applications are built to scale, and an elastic architecture that scales as the traffic grows ensures business needs are not impacted by a large customer base. The architecture should encompass scalability approaches in the design, code, and infrastructure phases.

Deployment Strategy

A deployment process, whether in the cloud or on-premises, should be an integral part of the architecture design. Deployment methodologies such as continuous integration and continuous deployment (CI/CD) should ideally be a fabric of this design to streamline the deployments of builds.

Security

In today's world of ubiquitous and pervasive computing, a user's sensitive information and overall data security is of paramount importance. An architectural design should insist on incorporating security procedures as a pattern and enforce strong security practices via configuration or convention.

User Experience and Inclusivity

Pertinent to user-facing systems, the end-user experience is paramount in architecture design. Experience architecture (XA) is the process of articulating the user's journey from one subsystem to another within an application, and is vital in providing the user with helpful controls, hints, and other methods to navigate. The system architecture should also include accessibility design as a part of the user experience, so they can navigate an application thoroughly regardless of physical or cognitive differences.

Recovery and Planning

Data recovery (DR) and business continuity planning (BCP) should be vital parts of an architectural design that ensures business needs are not largely affected when an unforeseen event occurs.

Unit Testing

A resilient architecture should incorporate unit testing as an essential component of its design. A code coverage report generated on each build provides opportunities for code reviews within the team where any inconsistencies can be discovered quickly. Automation should be explored as an integral element of the architecture wherever possible, and not as an afterthought.

Application Performance Monitoring

Even the best engineered systems fail. And when they do, the architecture should be robust enough to offer the end-users and the development teams support information with what went wrong, when, and why. Application performance monitors (APMs) are particularly useful in providing detailed insights on application issues.

Overall, a system architect's role and performance is defined by the concept, design, development, and maintenance of the application they architect.