Electronic Circuit CAD

© David Norris

- Delivery: 90 minutes, 14:00 – 15:50, 10 minute break, 14:55 – 09:05 Beijing time, Monday – Friday 29/05/2023 – 09:06/2023
- Learning Outcomes:

On successful completion of this module a student will be able to:

1 Demonstrate understanding of Concepts and Applications of Computer Graphics and Geometric Modelling.

2 Apply 3D modelling principles and methods to develop computer based designs.

3 Evaluate CAD data for different applications in Design and Manufacturing lifecycle.

4 Create full design documentation including technical drawings and quality management principles in systems engineering.

How to obtain these notes

- Download from

http://dfdn.info/teaching/electronic-circuits-cad/

Indicative content

• Introduction of Computer Graphics and Geometrical Modeling

• Complex sketches including 2D and 3D

• Feature-based modelling and use of feature library.

• Assembly modelling of complex products and use of library parts.

• Advanced Modeling: Surface modelling, Generative Modeling.

• Customising of CAD systems using macros and Application

Programming Interface (API)

• Applications and Analysis of CAD in Product Life Cycle:

CAD/CAM/CNC/CAE, and data transfer

• Design Documentation: Technical Drawings, Animation and Simulations

• Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

• Address challenges of information security in product lifecycle management

Teaching and learning activity

- The Module will include lectures and guided practical exercises
- and projects that will allow the creation and optimisation of CAD
- models, assemblies, and manufacturing drawings demonstrating
- the competence and understanding achieved by the student of the

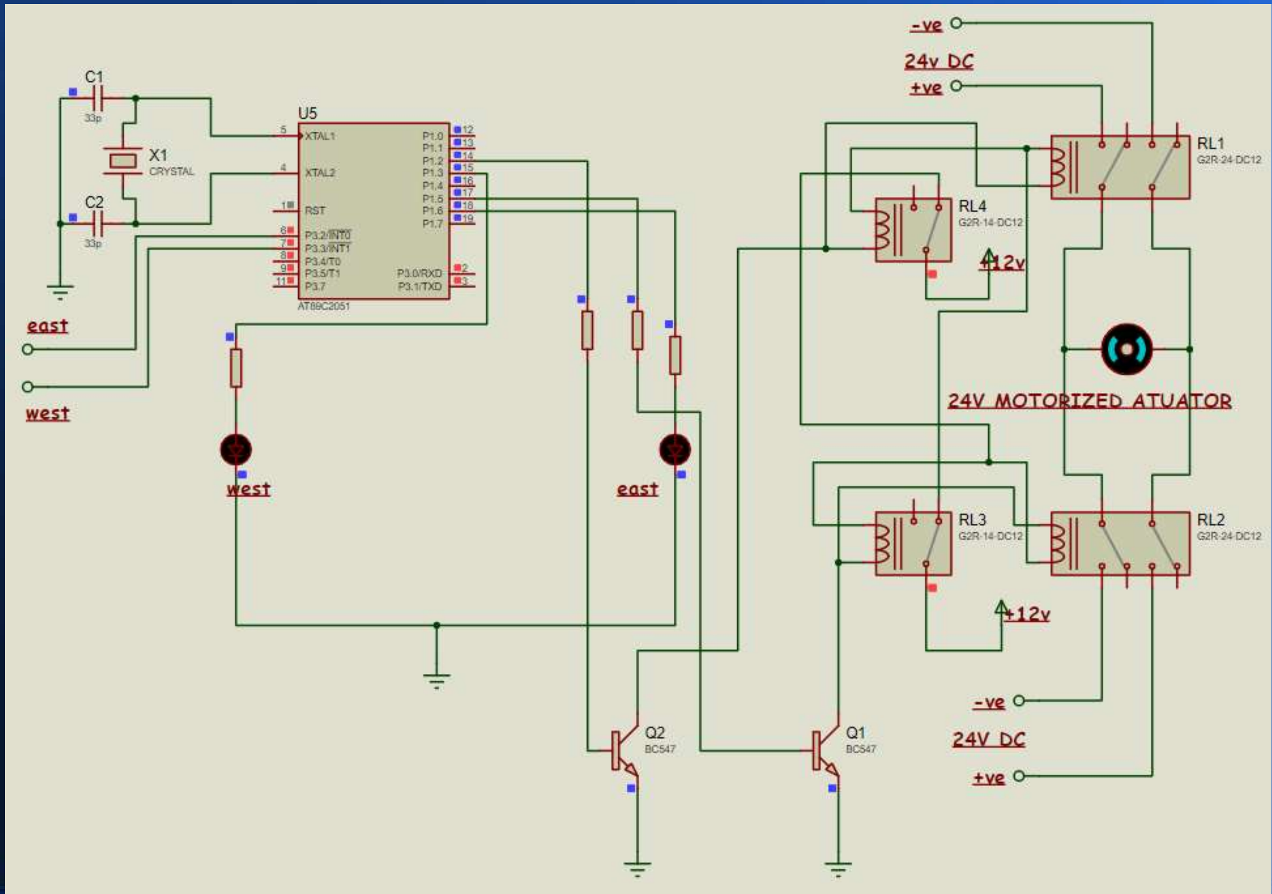
main content.

About me

- I am a British man and native English Speaker.
- I have a first degree in Electronics (Bsc Electronics) and a masters' degree in Computer Science (Msc Computer Science)
- I currently live in London, UK but I will be relocating to Ouagadougu, Burkina Faso in 2023
- I currently teach English as a foreign language, as well as electronics, mathematics and physics
- I have taught another group similar to this one
- Any questions? Feel free to ask!

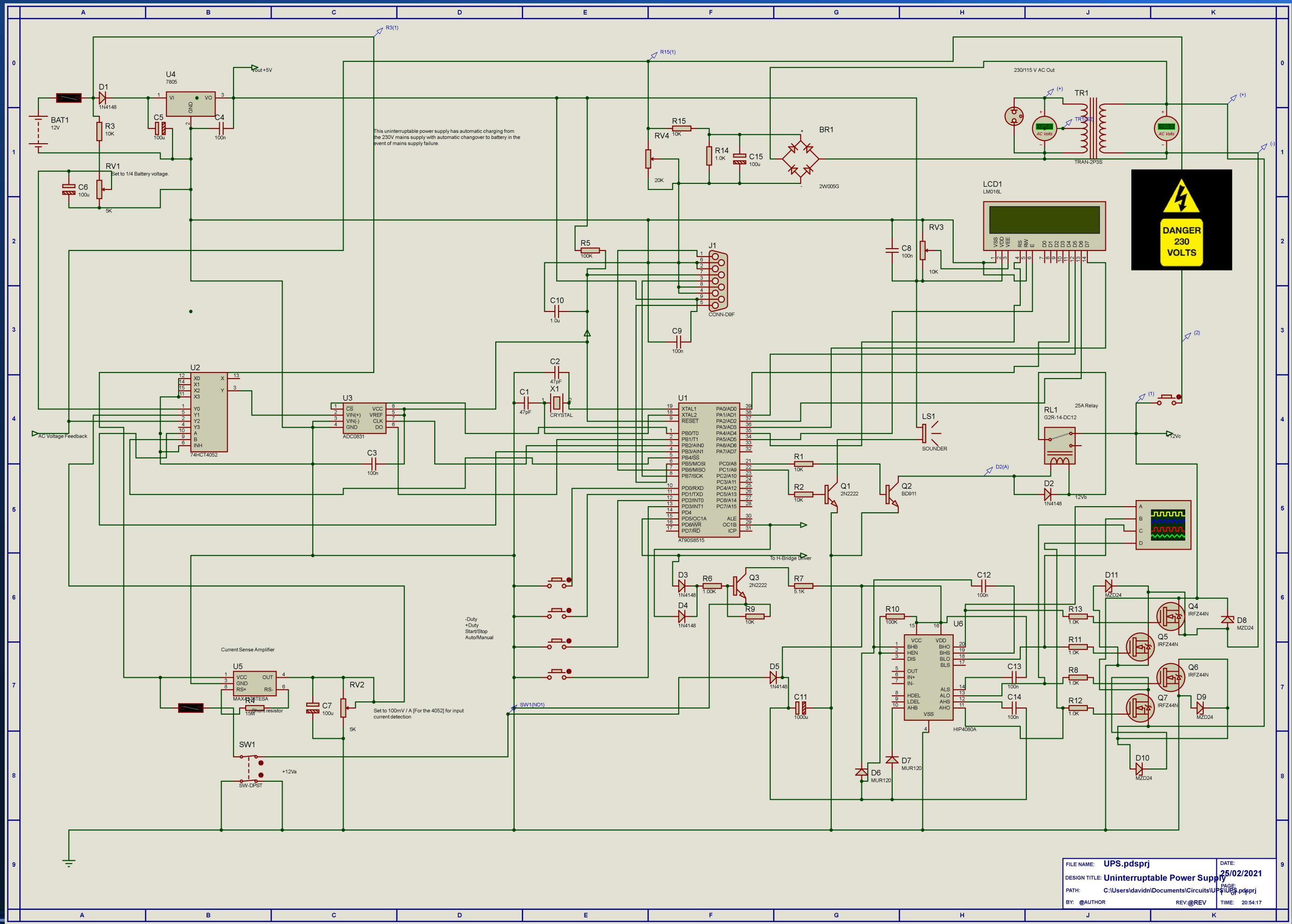
Electronic Circuit CAD [Computer Aided Design]

- A simplified version of a device for keeping solar pannels aimed at the sun. I intend to set up a solar powered electronic business in Western Africa in the medium term. Used a mictocontroller and stepper motors. Firmware is written in 'C' and can use a time based tracking approach or a maximum power point tracking approach. The power yield of a solar panel is proportional to the cosine of the incident angle.



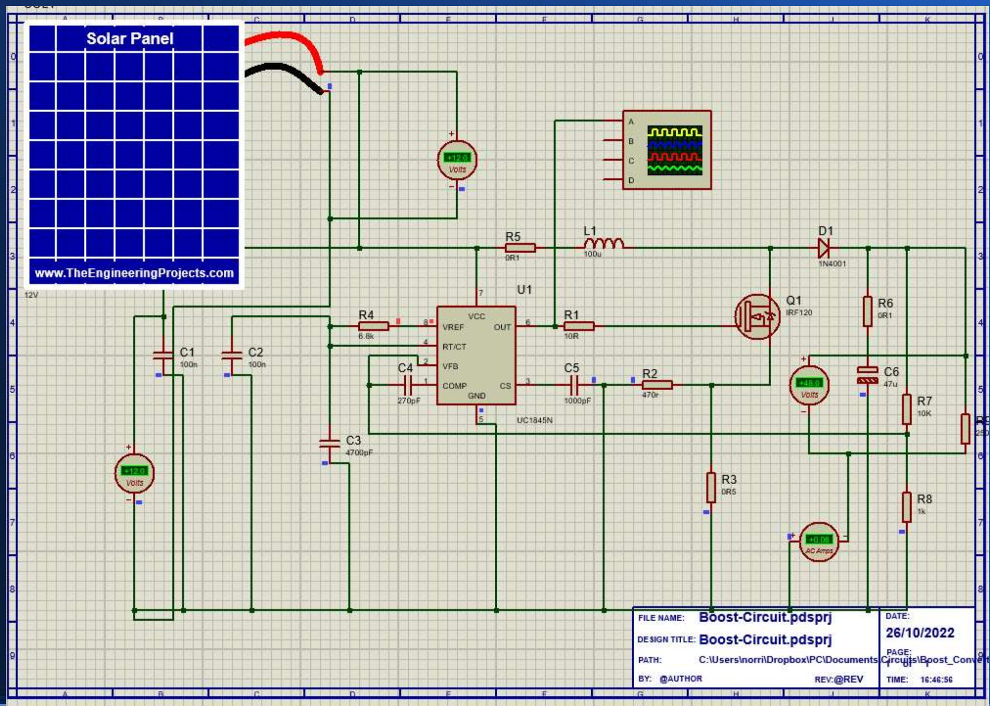
Electronic Circuit CAD [Computer Aided Design]

- A low cost Uninteruptable power supply design for use in Western Africa, where I intend to establish a Business. Uses a microcontroller, firmware written in 'C'. Copyright © David Norris, 2021



Electronic Circuit CAD [Computer Aided Design]

- A boost converter. Before semiconductors, the only way to change DC voltages was to use an inefficient motor/generator. Transformers operate only on AC supplies, and long distance distribution needs high voltage to reduce ohmic cable losses.



Introduction of Computer Graphics and Geometrical Modeling

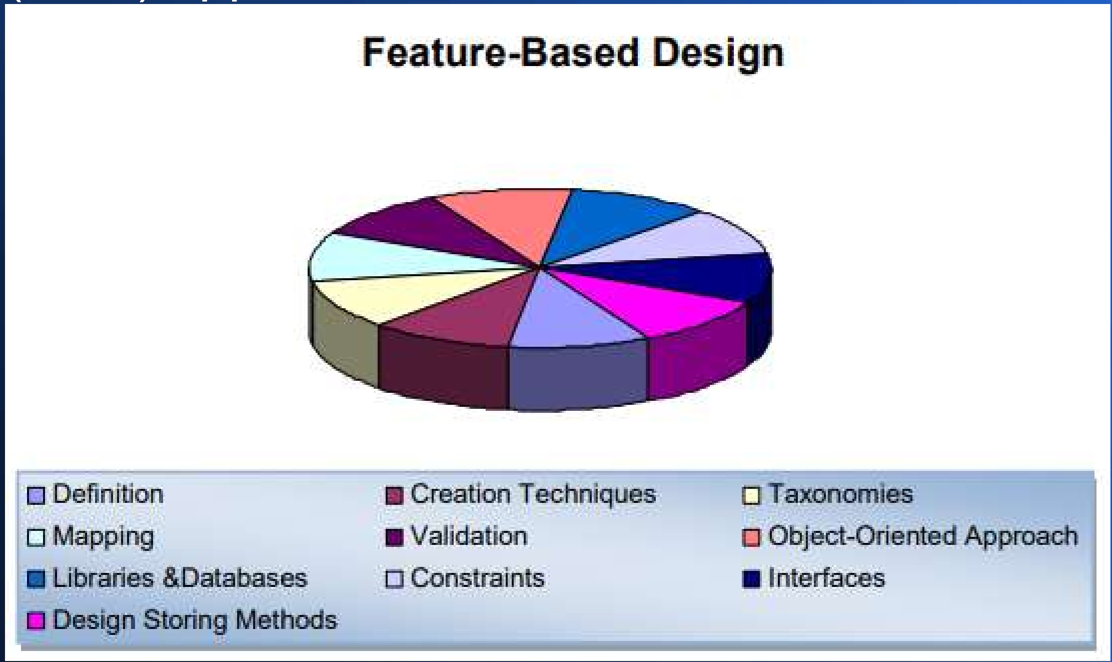
- Today CAD workstations are an essential tool for design engineer. Indeed, over the last 40 years, computers have revolutionised the way that engineering designs are created. Initially, computers were used as a drafting tool. Progressively, the technological advances in computer software and hardware introduced the computer as a medium to model engineering creations in three-dimensions.
- "Geometric modeling" became a powerful technique in engineering, providing the ability to determine through different types of modelers the construction sequences, behavior, strengths and weaknesses, and other characteristics of most of the engineering models before they were physically constructed. The next step was FBM (Feature–Based Modeling). Features were introduced in the engineering designers' arsenal of tools that helped bring together design with manufacturing and provided a means for engineers to develop a link between CAD and CAM (Computer-Aided Manufacturing) as well as techniques to automate the procedures.

Introduction of Computer Graphics and Geometrical Modeling

- This automation resulted in higher production rates (efficiency) and capital cost reduction. With the aid of modern, state-of-the-art design software, the creation of a part's manufacturing code (i.e. Numerical Control-NC Code) can now be made on the fly. However, it is still a relatively new field and the future success of such computer software packages will depend on the ability of CAD systems to implement new techniques, handle more complexity, and potentially aid in the decision-making in order to ultimately make the design and representation of a product a more accurate and efficient process. This lead in turn to feature based design, FBD.
- The next slide illustrates the different areas of research within FBD. This paper presents an overview of the research carried out in feature-based design (FBD). Current methods, definitions and procedures are reviewed and analyzed further so as to determine their drawbacks and to emphasize the need for development of new techniques.

Introduction of Computer Graphics and Geometrical Modeling

- There have been various different definitions about what exactly a feature represents. Early informal feature definitions were analyses and process planning oriented. That was because at the time most of the Computer-Aided Design (CAD) applications needed a boost so as to increase the



FEATURE CREATION TECHNIQUES

- design efficiency with the integration of other applications based on analyses and process planning. An early feature definition was “a specific geometric configuration formed on the Computer-Aided Design & Applications, surface, edge or corner of a workpiece, intended to modify outward appearances or to assist in achieving a given function.
- Design features are usually defined as sets of geometric entities that represent certain shapes patterns and have certain functions or embedded information. Manufacturing features are usually portions of a workpiece that can be generated with metal removal processes. In general though, a feature is a physical constituent of a part and as a result parts are physical constituents of assemblies. Thus, any feature attribute is a characterisation of parts and their assemblies.

FEATURE CREATION TECHNIQUES

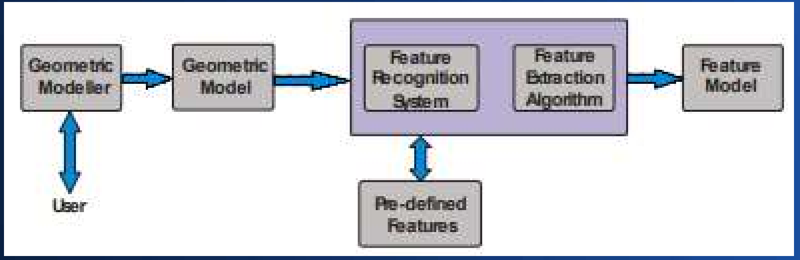
- There are three major design and manufacturing strategies/approaches whereby Feature-Based Design can be used. The first approach is called Concurrent Design. In concurrent design the design engineer creates models with respect to their manufacturing processes. As such, there is an optimisation in the production period that helps the engineers to think in advance and be able to modify the original model in case of any possible inadequacies by following a path of iterative procedures. It can be stated that this strategy is mainly process planning based and its main characteristics are as follows.
- • Design is carried out in manufacturing mode
- • The primitives that are used are mainly manufacturing features
- • It is suitable for fabricating moulds on matching centres for small batches.

FEATURE CREATION TECHNIQUES

- The second approach is called Design for Assembly (DFA). This approach has become one of the favourite in the recent years because of its simplified use. Most of the current DFA software packages are interactive and user friendly. The main target of this approach is to minimise the possible number of parts in an assembly and at the same time minimise lead time, reduce the capital and running costs. These techniques are primarily quantitative or qualitative and use the cost-tolerance design methodology normally.
- Finally, the third is the parametric design which is the approach and focus of this paper. It groups similar products into product families helping in that way the designer to create a new or modify pre-existing solid models on the fly. Thus new, and reuse of previous designs can be achieved with a variant approach. This approach has emerged from the integration of CAD with computer-aided process planning (CAPP). Today, it is widely used by the variety of designers and software developers such as Parametric Technology Corporation (PTC), SolidWorks, Unigraphics, Aries, AutoDesk and others. Including FreeCAD.

Form-Feature Recognition

- Often referred to as Automatic Feature Definition (AFD), this approach was mainly established to fully automate the feature creation or extraction procedure. Features are automatically recognised from an object subjected to consideration [40]. Gindy [22] defined form features as volumes enveloped by entry/exit and depth boundaries and proposed a hierarchical structure for form features classification. Based on the different categories, form features are classified into classes and sub-classes. There are many different classification schemes.



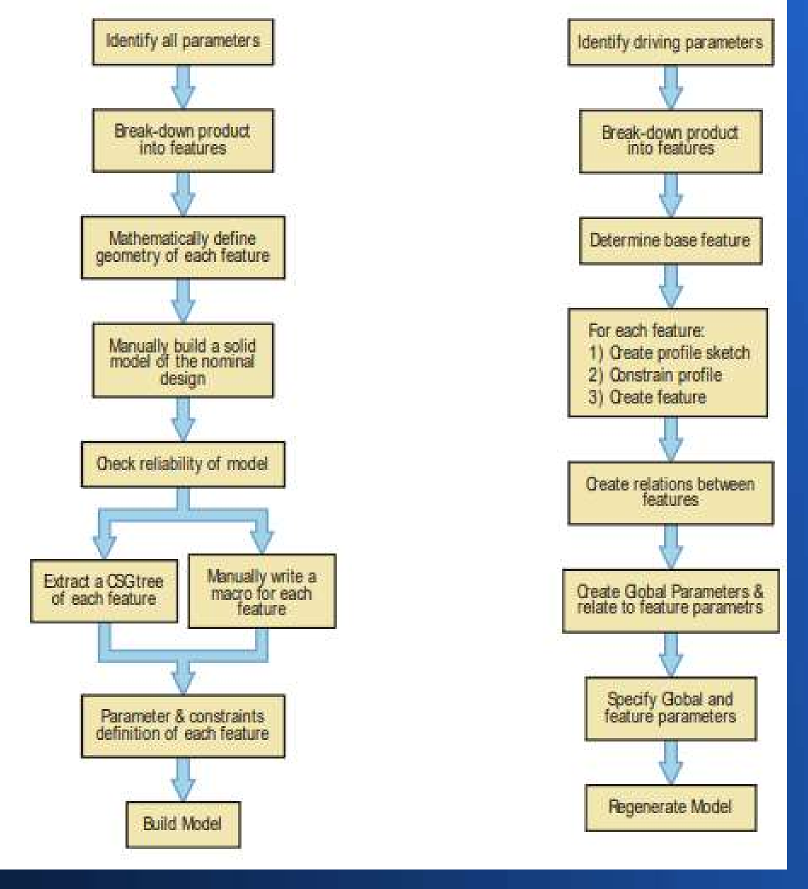
Design by Features (DbF)

- This is a rather manual process in which the user builds a feature-based (geometric) model with the aid of feature libraries. In those libraries, a variety of predefined features are stored in a form accessible by the user through a solid modelling software package. The user can specify dimensions, location parameters and various other attributes and relationships so as to position the feature in the desired location. This can be an advantage to the user interms of reducing the design lead-time. On the other hand, it is possible to experience difficulties due to insufficient feature entries in the feature library.

FEATURE-BASED DESIGN REUSE

- It is essential to be able to store engineering designs that can be used as a reference for the future. For centuries, engineers have relied on past designs to develop and adapt new ones. Semi-automated methods for capturing detailed designs are an attempt to incorporate the advantages of retaining a high level on design intent, whilst using techniques, such as Parametric and Variational Design and Feature Based Design, to automate the generation (or instancing) of similar designs, i.e. its variants.
- The two principal techniques for the semi-automated capture of past (and the creation of new) designs, are the Generative (sometimes called Procedural) and Variant Design Methods illustrated next.

FEATURE-BASED DESIGN REUSE



Introduction of Computer Graphics and Geometrical Modeling

- Computer Aided Engineering (CAE) is a vast subject that comprises of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). CAD can be further sub divided into Finite Element Analysis, Computer Aided Drafting and Computer Geometric Modelling.
- Computer geometric modelling is the mathematical representation of an object's geometry using software. A geometric model contains description of the modelled object's shape. Since geometric shapes are described by surfaces, curves are used to construct them. Computer geometric modelling uses curves to control the object's surfaces as they are easy to manipulate. The curves may be constructed using analytic functions, a set of points, or other curves and surfaces. There are number of software programs that then allow the mathematical description of the object to be displayed as an image on the monitor.

Introduction of Computer Graphics and Geometrical Modeling

- Computer geometric modelling is the field that discusses the mathematical methods behind the modelling of realistic objects for computer graphics and computer aided design. It uses computers to store the data and the geometric properties of a system / part.

Introduction of Computer Graphics and Geometrical Modeling

- A geometric model of an object can be created using these 3 steps:
- Create basic geometric object using the commands like points, line and circles in CAD software
- Use commands like achieve scaling, rotation, etc. to transform these geometric elements
- Integrate the various elements of the object to form the final geometric model
- A Bit of History…
- In engineering, modelling refers to the way data are represented in the computer memory and the way in which it is visualised. The data is stored in a suitable data structure that is easy to access for the visualisation algorithm. Geometric modelling is the process of capturing the properties of an object or a system using mathematical formulae. Computer geometric modelling is the field that discusses the mathematical methods behind the modelling of realistic objects for computer graphics and computer aided design. It uses computers to store the data and the geometric properties of a system / part. It is a relatively new technology that has expanded rapidly thanks to advances in computer technology.

Introduction of Computer Graphics and Geometrical Modeling

- It is a relatively new technology that has expanded rapidly thanks to advances in computer technology. The 1st generation CAD programs of the 50’s were mostly non interactive. CAD users were required to create program codes to generate the desired 2D geometric shapes. Interactive CAD programs began to take shape in the 60’s and were mostly used in the automotive and the aerospace industry. The 60’s also marked the beginning of the development of finite element analysis methods for computer stress analysis and CAM for generating machine tool paths. The advent of microprocessors in the late 70’s boosted computing power of the computers. It also triggered the development of 3D CAD programs that were user friendly and most importantly, interactive. The 80’s saw the expansion of computer application software technology from very simple computer aided drafting to very complex computer aided design. It was at this time that 2D and 3D wire frames were introduced in the industry for geometric modelling as a tool to increase productivity.

Introduction of Computer Graphics and Geometrical Modeling

- During the process of geometric modeling the computer converts various commands given from within the CAD software into mathematical models, stores them as files and finally displays them as an image. The geometric models created by the designer can open at any time for reviewing, editing or analysis.
- CAD really came on its own with the introduction of 3D solid modelling technology, which boosted the usage of CAE technology in industry. The development of 3D modelling schemes started with 3D wireframes. It was a major leap in computer geometric modeling as it allowed designers to use a single object and view it from multiple angles. However, wire frames are susceptible to ambiguity because surface definition is not part of a 3D wireframe model. To overcome this lacuna, computer geometric modelling with surface definitions was the next logical step forward. With enhanced computer CPU power, the next enhancement in geometric modelling was solid modelling. Surface modelling organises and groups edges that define polygonal surfaces, making it easier for designers to visualise an object.

Introduction of Computer Graphics and Geometrical Modeling

- Solid Geometric Modelling:
- Solid modelling allows definition of an object's nodes, edges and surfaces. It is therefore a complete and unambiguous mathematical representation of a precisely enclosed and filled volume. Unlike the surface modelling method, solid modellers start with a solid or use topology rules to guarantee that all surfaces are stitched together properly.
- There are two predominant methods for representing solid models – constructive solid geometry and boundary representation. Constructive solid geometry combines basic solid objects (for example a rectangular prism, cylinder, cone, sphere, etc.). These shapes are simply added or deleted in order to form the final solid shape. In boundary representation, objects are defined in terms of their spatial boundaries. It defines the points, edges, surfaces of a volume, and / or issues commands that sweep or rotate a defined face into a third dimension to form a solid. The object is then made up of the union of these surfaces that completely and precisely enclose a volume.
-

Introduction of Computer Graphics and Geometrical Modeling

- The 80’s also saw a new paradigm called concurrent engineering. In this concept, the entire production team (designers, analysts, engineers, testers and production managers) work concurrently right from the inception of the project. Feature based parametric modelling is the result of this collaboration; we have covered it separately.
- To summarise, these are the various types of computer geometric modeling techniques as of date:
- Wire frame models (describe an object using boundary lines)
- Surface models (describe an object using boundary surfaces)
- Solid models (describe an object as a solid)

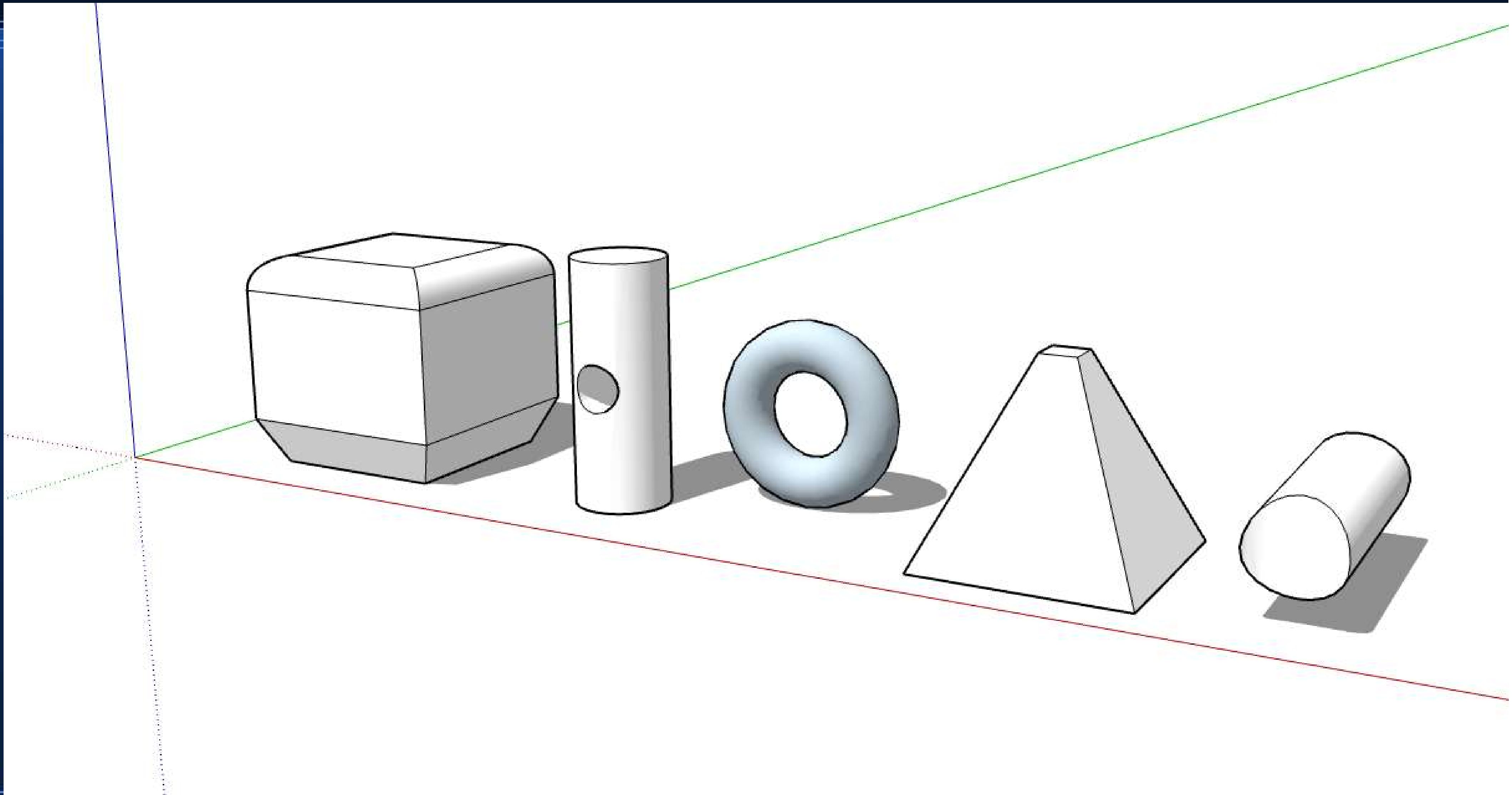
Introduction of Computer Graphics and Geometrical Modeling

- The Kernel and its Importance:
- Solid geometric modelling has grown in popularity. It is one of the most important applications of CAD. CAD programs that use solid modelling concept help design engineers to view the part / object as if it was really manufactured. The CAD software can even change the perspective and viewing angles. SolidWorks, CATIA and PTC Creo are a few popular tools used for computer geometric modelling – especially solid modelling.
- The heart of computer geometric modelling software is the kernel. Called either a geometric modelling kernel or solid modelling kernel, it is the code that determines how the image you see on the screen is actually described with math. Kernels are important because they are mathematical models of real and theoretical objects. Describing shapes in a mathematical representation involves making choices about how each shape is calculated and stored. Needless to say, the better a kernel, the better a geometric modelling software. In addition to all the functions required by programmers, a good kernel should also be reliable and quick. As the quality of a kernel affects the quality of the entire CAD system companies like PTC, SolidWorks, CATIA and others constantly strive to improve it.

Introduction of Computer Graphics and Geometrical Modeling

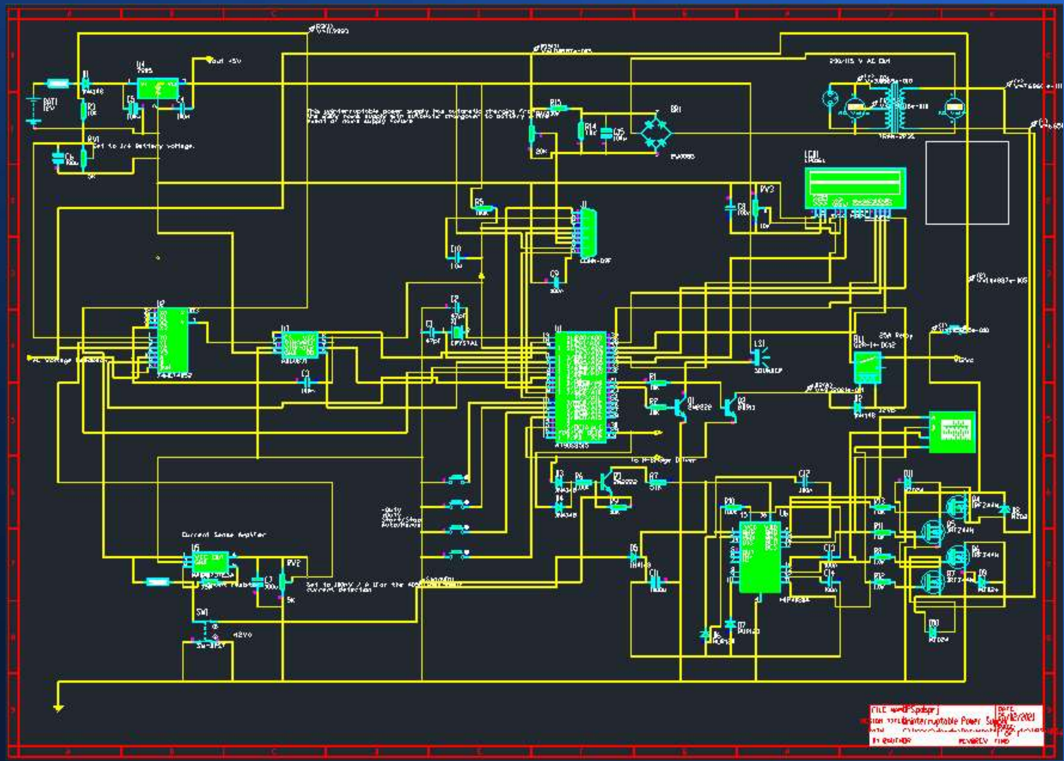
- Computer Geometric Modelling – the Future:
- Computer geometric modelling finds use in numerous sectors like industrial engineering, automotive engineering, aerospace, defence, power, etc. In addition, geometric modelling is also finding uses in many newer industries like robotics, medical imaging, visualisation, etc. Developing countries (like India, China and Brazil for example) have seen a tremendous growth in the demand for companies that provide geometric modelling – solid as well as feature based. This is because simulation is typically time consuming. Unless you have state of the art software like PTC Creo, SolidWorks or CATIA - which small companies typically do not have - it is tedious as well. There have been cases where the geometric modelling time itself took about half the PLM time due to the complexity of the geometry! The more intricate the surface details (surface mesh), the better it is for design engineers. Companies that provide geometric modeling as a service are therefore growing in popularity in India and elsewhere.

Complex sketches including 2D and 3D



Complex sketches including 2D and 3D

- This is a 2D representation of one of my own designs I plan to market in West Africa. It is an uninterruptable power supply, UPS.



THE ROLE OF CAD IN ELECTRONIC DESIGN AND MANUFACTURE

- Computer Aided Design (CAD) refers to any software that aids developers in the creation of their product. ECAD (Electronic Computer Aided Design) and EDA (Electronic Design Automation) refer specifically to CAD software designed for electronic design.. In the electrical engineering field, CAD has become pivotal to the ongoing process of manufacturing complex electronics like printed circuit boards (PCBs) and microprocessors that are designed down to the subatomic level. The software provides a large number of tools and functionality that allow electronics companies to make the products that drive the age of information technology. The main benefit is that designs are converted directly to a production machines format, removing the possibility of human error. This improves quality and keeps costs down. This is especially true on larger scale designs or projects.

The Role of ECAD

- Until an electronic device goes to the production line, ECAD is the most important tool of the design process. It performs the primary tasks of constructing electrical schematics, performing simulations, and creating physical blueprints for electronic devices ranging from the latest microprocessor to powerful graphic processing units. It is the primary reason that electronics have advanced from processors with just over 2000 transistors to ones with billions packed onto them.

ECAD Functionality

- The available functionality of ECAD software can be summed up as the design and testing of complex electronics schematics on both the physical and logical level. The schematics for both aspects are generated are stored in digital format with frequent backups, making it easier to modify and distribute.
- On the electrical side, ECAD programs typically come with the ability to design the electronics schematics from the ground up and eventually run simulations to determine how the hardware will respond when an electronic input is given. This is done by using the software language to replicate the logical functions that electronics goes through. For a simple example, if an AND logic gate is shown on the schematic, the simulation would take all the inputs into the gate and perform the same process at the application level. The program would do this for all possible inputs in order to quickly and automatically test the schematic. Many of the programs also offer the ability to create “cells” that act similarly to logic gates or solid-state chips.

ECAD Functionality

- The available functionality of ECAD software can be summed up as the design and testing of complex electronics schematics on both the physical and logical level. The schematics for both aspects are generated are stored in digital format with frequent backups, making it easier to modify and distribute.
- On the electrical side, ECAD programs typically come with the ability to design the electronics schematics from the ground up and eventually run simulations to determine how the hardware will respond when an electronic input is given. This is done by using the software language to replicate the logical functions that electronics goes through. For a simple example, if an AND logic gate is shown on the schematic, the simulation would take all the inputs into the gate and perform the same process at the application level. The program would do this for all possible inputs in order to quickly and automatically test the schematic. Many of the programs also offer the ability to create “cells” that act similarly to logic gates or solid-state chips.

ECAD Functionality

- Every electrical component has a physical manifestation, and complete ECAD software manages this aspect as thoroughly as it cares for the logic process. The software allows the engineers to place the physical components represented in the electrical diagram onto a model of its physical form. Any discrepancies, such as overlapping parts or a lack of room, are noted and can then be corrected well before the manufacturing process.

ECAD Programs

- There are a number of both open source and commercial options available for EDA software. On the commercial side, the top options include programs such as: Zuken’s CADSTAR, Mentor Graphics’ collection of programs, Labcentre's Proteus, and Synopsis Design Compiler. The most popular open source tool is gEDA, developed for Linux due to the low availability of quality open source EDA tools. I have the most experience with Proteus.

Technical Drawing

- Firstly let us look briefly at the old fashioned way!
- Technical drawing, drafting or drawing, is the act and discipline of composing drawings that visually communicate how something functions or is constructed.
- Technical drawing is essential for communicating ideas in industry and engineering. To make the drawings easier to understand, people use familiar symbols, perspectives, units of measurement, notation systems, visual styles, and page layout. Together, such conventions constitute a visual language and help to ensure that the drawing is unambiguous and relatively easy to understand. Many of the symbols and principles of technical drawing are codified in an international standard called ISO 128.
- The need for precise communication in the preparation of a functional document distinguishes technical drawing from the expressive drawing of the visual arts. Artistic drawings are subjectively interpreted; their meanings are multiply determined. Technical drawings are understood to have one intended meaning.

Design Documentation: Technical Drawings, Animation and Simulations

- A drafter, draftsperson, or draughtsman is a person who makes a drawing (technical or expressive). A professional drafter who makes technical drawings is sometimes called a drafting technician.
- Sketching:
- A sketch is a quickly executed, freehand drawing that is usually not intended as a finished work. In general, sketching is a quick way to record an idea for later use. Architect's sketches primarily serve as a way to try out different ideas and establish a composition before a more finished work, especially when the finished work is expensive and time-consuming.
- Architectural sketches, for example, are a kind of diagrams.These sketches, like metaphors, are used by architects as a means of communication in aiding design collaboration. This tool helps architects to abstract attributes of hypothetical provisional design solutions and summarise their complex patterns, thereby enhancing the design process.

Design Documentation: Technical Drawings, Animation and Simulations

- The basic drafting procedure is to place a piece of paper (or other material) on a smooth surface with right-angle corners and straight sides—typically a drawing board. A sliding straightedge known as a T-square is then placed on one of the sides, allowing it to be slid across the side of the table, and over the surface of the paper.
- "Parallel lines" can be drawn simply by moving the T-square and running a pencil or technical pen along the T-square's edge. The T-square is used to hold other devices such as set squares or triangles.
- In this case, the drafter places one or more triangles of known angles on the T-square—which is itself at right angles to the edge of the table—and can then draw lines at any chosen angle to others on the page. Modern drafting tables come equipped with a drafting machine that is supported on both sides of the table to slide over a large piece of paper. Because it is secured on both sides, lines drawn along the edge are guaranteed to be parallel.

Technical Drawing

- In addition, the drafter uses several technical drawing tools to draw curves and circles. Primary among these are the compasses, used for drawing simple arcs and circles, and the French curve, for drawing curves. A spline is a rubber coated articulated metal that can be manually bent to most curves.
- Drafting templates assist the drafter with creating recurring objects in a drawing without having to reproduce the object from scratch every time. This is especially useful when using common symbols; i.e. in the context of stagecraft, a lighting designer will draw from the USITT standard library of lighting fixture symbols to indicate the position of a common fixture across multiple positions. Templates are sold commercially by a number of vendors, usually customised to a specific task, but it is also not uncommon for a drafter to create his own templates.

Technical Drawing

- This basic drafting system requires an accurate table and constant attention to the positioning of the tools. A common error is to allow the triangles to push the top of the T-square down slightly, thereby throwing off all angles. Even tasks as simple as drawing two angled lines meeting at a point require a number of moves of the T-square and triangles, and in general, drafting can be a time-consuming process.
- A solution to these problems was the introduction of the mechanical "drafting machine", an application of the pantograph (sometimes referred to incorrectly as a "pentagraph" in these situations) which allowed the drafter to have an accurate right angle at any point on the page quite quickly. These machines often included the ability to change the angle, thereby removing the need for the triangles as well.

Technical Drawing

- n addition to the mastery of the mechanics of drawing lines, arcs and circles (and text) onto a piece of paper—with respect to the detailing of physical objects—the drafting effort requires a thorough understanding of geometry, trigonometry and spatial comprehension, and in all cases demands precision and accuracy, and attention to detail of high order.
- Although drafting is sometimes accomplished by a project engineer, architect, or shop personnel (such as a machinist), skilled drafters (and/or designers) usually accomplish the task, and are always in demand to some degree.

Technical Drawing – the modern approach

- Today, the mechanics of the drafting task have largely been automated and accelerated through the use of computer-aided design systems (CAD).
- There are two types of computer-aided design systems used for the production of technical drawings: two dimensions ("2D") and three dimensions ("3D").
- An example of a drawing drafted in AutoCAD
- 2D CAD systems such as AutoCAD or MicroStation replace the paper drawing discipline. The lines, circles, arcs, and curves are created within the software. It is down to the technical drawing skill of the user to produce the drawing. There is still much scope for error in the drawing when producing first and third angle orthographic projections, auxiliary projections and cross-section views. A 2D CAD system is merely an electronic drawing board. Its greatest strength over direct to paper technical drawing is in the making of revisions. Whereas in a conventional hand drawn technical drawing, if a mistake is found, or a modification is required, a new drawing must be made from scratch, the 2D CAD system allows a copy of the original to be modified, saving considerable time. 2D CAD systems can be used to create plans for large projects such as buildings and aircraft but provide no way to check the various components will fit together.

Technical Drawing

- A 3D CAD system (such as KeyCreator, Autodesk Inventor, or SolidWorks) first produces the geometry of the part; the technical drawing comes from user defined views of that geometry. Any orthographic, projected or sectioned view is created by the software. There is no scope for error in the production of these views. The main scope for error comes in setting the parameter of first or third angle projection and displaying the relevant symbol on the technical drawing. 3D CAD allows individual parts to be assembled together to represent the final product. Buildings, aircraft, ships, and cars are modeled, assembled, and checked in 3D before technical drawings are released for manufacture.
- Both 2D and 3D CAD systems can be used to produce technical drawings for any discipline. The various disciplines (electrical, electronic, pneumatic, hydraulic, etc.) have industry recognised symbols to represent common components.

Design Documentation: Technical Drawings, Animation and Simulations

- BS and ISO produce standards to show recommended practices but it is up to individuals to produce the drawings to a standard. There is no definitive standard for layout or style. The only standard across engineering workshop drawings is in the creation of orthographic projections and cross-section views.
- In representing complex, three-dimensional objects in two-dimensional drawings, the objects can be described by at least one view plus material thickness note, 2, 3 or as many views and sections that are required to show all features of object.

Design Documentation: Technical Drawings, Animation and Simulations

- Here is a typical work flow for design review, once the 3-D modeling is complete:
- Create an exploded view of the 3-D model.
- Create 2-D illustrations, complete with BOMs and balloons.
- Create animations and publish 3-D content.
- Use the 3-D model to create illustrations and interactive content for end-user manuals and training.
- A sketching tool—Exploded Line Sketch—is useful for adding the connecting lines between the exploded components. Those connecting lines automatically hide when the view is collapsed.
- Those explode steps determine how the components are positioned relative to each other. The sequence of those steps controls when the move happens. After the explode steps are created, their order of appearance in the list can be changed with mouse drag.

Design Documentation: Technical Drawings, Animation and Simulations

- Some labor is expended in preparing an exploded feature. The skills required for creating an exploded view are not required for invention and 3-D modeling. However, several useful results are derived from the effort that goes into an exploded feature:
- Demonstration of instant toggling between collapsed and exploded views using the CAD workstation
- Self-paced review of the design using the eDrawings® Viewer software, which plays the explode/collapse on demand
- Preparation of fabrication and assembly instructions (for example, inserting the exploded view into a drawing)
- Importing the exploded view into interactive documents composed with dedicated authoring software
- Creating animations and AVI movies of how the components fit together

Design Documentation: Technical Drawings, Animation and Simulations

- Instant Animation
- The Animate tool's controller—Figure 3b—presents options to loop, reciprocate, and otherwise play. It moves the components according to the sequence of explode steps in the explode feature.
- CAD tip: It is easy to change the sequence of explode steps using mouse drag-and-drop.
- The Animate tool will also typically export to an AVI movie file. Here is a movie director's tip: Create a visual simulation, smooth and graceful. Allow the audience to watch the explode-to-collapse sequence a few times; use subtle changes in point of view to add interest; and end with the components in their starting position.
- I also use Camtasia Studio to produce visual demonstrations; there are a number of these on my website. For example, this is a deminstration of a teaching circuit made with Camtasia Studio: http://dfdn.info/video/Photovoltaic-3Phase/Photovoltaic-3Phase.html

Design Documentation: Technical Drawings, Animation and Simulations

- Instant Animation
- The Animate tool's controller—Figure 3b—presents options to loop, reciprocate, and otherwise play. It moves the components according to the sequence of explode steps in the explode feature.
- CAD tip: It is easy to change the sequence of explode steps using mouse drag-and-drop.
- The Animate tool will also typically export to an AVI movie file. Here is a movie director's tip: Create a visual simulation, smooth and graceful. Allow the audience to watch the explode-to-collapse sequence a few times; use subtle changes in point of view to add interest; and end with the components in their starting position.
- I also use Camtasia Studio to produce visual demonstrations; there are a number of these on my website. For example, this is a deminstration of a teaching circuit made with Camtasia Studio: http://dfdn.info/video/Photovoltaic-3Phase/Photovoltaic-3Phase.html

Design Documentation: Technical Drawings, Animation and Simulations

- To sumarise:
- The purpose of CAD is to optimise and streamline the designer's workflow, increase productivity, improve the quality and level of detail in the design, improve documentation communications and often contribute toward a manufacturing design database. CAD software outputs come in the form of electronic files, which are then used accordingly for manufacturing processes.
- Compared to traditional technical sketching and manual drafting, the use of CAD design tools can have significant benefits for engineers and designers:
- Lower production costs for designs;
- Quicker project completion due to efficient workflow and design process;
- Changes can be made independent of other design details, without the need to completely re-do a sketch;
- Higher quality designs with documentation (such as angles, measurements, presets) built into the file;

Design Documentation: Technical Drawings, Animation and Simulations

- Clearer designs, better legibility and ease of interpretation by collaborators, as handmade drawings are not as clear or detailed;
- Use of digital files can make collaborating with colleagues more simple; and
- Software features can support generative design, solid modeling, and other technical functions.

Feature Based Modelling

- Feature based modeling is an indispensable tool for integrating design and manufacturing processes in CAPP system. Present work proposes a new approach to generate process plan from feature-based modeling, based on an integrated geometric modeling system that supports both feature-based modeling and information storage. This feature based modeling approach eliminates extraction of features from the programs of the already drawn components. As a result, feature information is directly available to downstream activities.
- The system comprises of three modules for different tasks such as feature-based modeling module, feature information storage module and process plan generation module. The proposed feature based modeling system provides a graphical environment for solid modeling in AutoCAD. The system is user friendly, flexible, expandable and reduces over all lead time. Present system is developed only for turning components and limited to selective machining features. To generalise the system, it can be extended to prismatic components and for more machining features. Generalised system can be used in manufacturing industry for automated process plan generation.

Feature Based Modelling

- Present manufacturing industry demands more automation from design stage to manufacturing stage. There is much interest in manufacturing firms to automate the task of process planning using computer-aided process planning (CAPP) systems. CAPP is usually considered to be part of Computer-Aided Manufacturing (CAM). However, this tends to imply that CAM is a stand-alone system. In fact, a synergy results when CAM is combined with computer-aided design to create a CAD/CAM system. In such a system, CAPP becomes thedirect connection between design and manufacturing. Process planning includes identification of the processes, machine tools, cutting tools, setups and fixtures to produce the desired product, along with geometric information. Even today, the activities of process planning are partially based on the skill of experienced process planners, which results in time-consuming procedures.

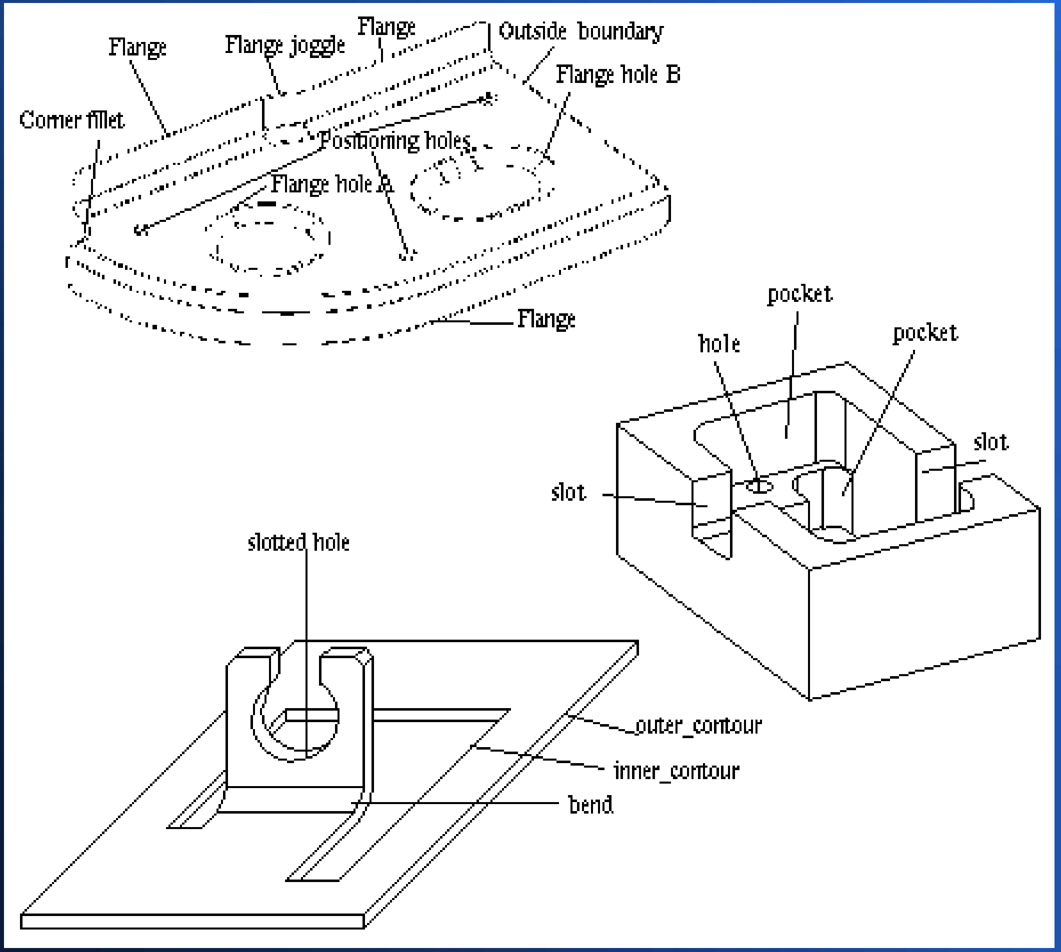
Feature Based Modelling

- As process planning is very complex, it would be desirable to use computer-aided approaches to relieve the process planner from routine activities andreduce the time and cost of the task. Because of the need to respond quickly to highly variable market demands, the development of computer-aided process planning (CAPP) systems is necessary.
- To achieve this computer integrated manufacturing (CIM) system, information about machining component is must. In terms of integrating CAPP systems, the feature-based approaches have been recognised as essential tools for eventually integrating process planning and design. Feature-based approaches are divided into two groups, namely, feature recognition and design by feature. The feature recognition approach examines the topology and geometry of a part and matches them with the appropriate definition of predefined features. The design by feature approach builds a part from predefined features where their attributes are attached.

Feature Based Modelling

- Many businesses use Autocad for their designwork. But Autocad is FAR from cheap ($2000 USD per year)! This is far too expensive for a student (or a teacher!), so I use a free alternative called FreeCAD. You can obtain this from my site here: http://dfdn.info/downloads for Windows, Linux or Macintosh.
- Feature based design:
- Features can be viewed upon as information sets that refer to aspects of form or other attributes of a part, in such a way that these sets can be used in reasoning about design, performance and manufacture of the part or the assemblies they constitute. The following slide shows some examples of features occurring on some different parts.

Feature Based Modelling



Feature Based Modelling

- Feature technology, is expected to be able to provide for an adequate basis for the integration of design and the subsequent applications such as engineering analysis, process planning, machining and inspection. Over the last few years, a vast number of papers and other publications on feature technology has come forward.
- Currently, three main views are discerned on how to obtain application features, such as manufacturing features, analysis and inspection features, from a product model.
- Feature recognition:
- In feature recognition, application features are automatically or interactively recognised from a model of the object under consideration. Product models from both solid modellers and feature based modellers can be subjected to feature recognition.Feature recognition is mainly used in CAPP systems; however, feature recognition may also be part of CAD functionality.

Feature Based Modelling

- Design by features
- A product model can be built by using (design) features; this is known as design by features or feature based modelling. One can start either with a more or less complete geometric model and define form features on it, or one starts from scratch by combining form features from a standard library. Design with pre-defined form features can reduce the number of input commands substantially. This is especially advantageous in re-design. The parametric representation of features provides a powerful way to change features with respect to their dimensions. Features can serve as functional elements to designers. Design features often differ from "downstream" application features. However downstream application features that are considered in this thesis are manufacturing features as used in process planning. The features are either similar to the "downstream" application features, in which case no special actions are required, or the design features differ from the manufacturing features in which case feature mapping or feature recognition are necessary. Mapping functionality has been identified as part of the CAD functionality while feature recognition has been identified as part of CAPP functionality.

Feature Based Modelling

- This has historic reasons. Initially (academic) feature based design systems were equipped with mapping functionality for linkage with downstream applications. Process planning systems were equipped with feature recognition modules to be able to transform CAD models into appropriate chunks of manufacturing information. Feature mapping need not necessarily be part of feature based design systems and equally feature recognition and feature identification need not necessarily be part of process planning systems.
- Interactive feature definition/feature identification
- In this approach, features are "defined" interactively. Most often, this is carried out by identifying the faces belonging to a certain feature on the product model that is under consideration. The feature that has been identified in this way, often is not generic in the sense that after having been identified, it can automatically be recognised from another product model. Therefore, in this thesis, the term feature identification is used in two different contexts. It can be used for the process of identifying the elements of a feature as a part of a product model without the possibility to reuse this information on other product models. It can also be used as a part of more generic feature definition. Feature definition is defined here as the definition of generic features, which can be used over and over again, either in CAD or in CAPP or even in both CAD and CAPP.

Feature Based Modelling

- Method for obtaining manufacturing features – flowchart.

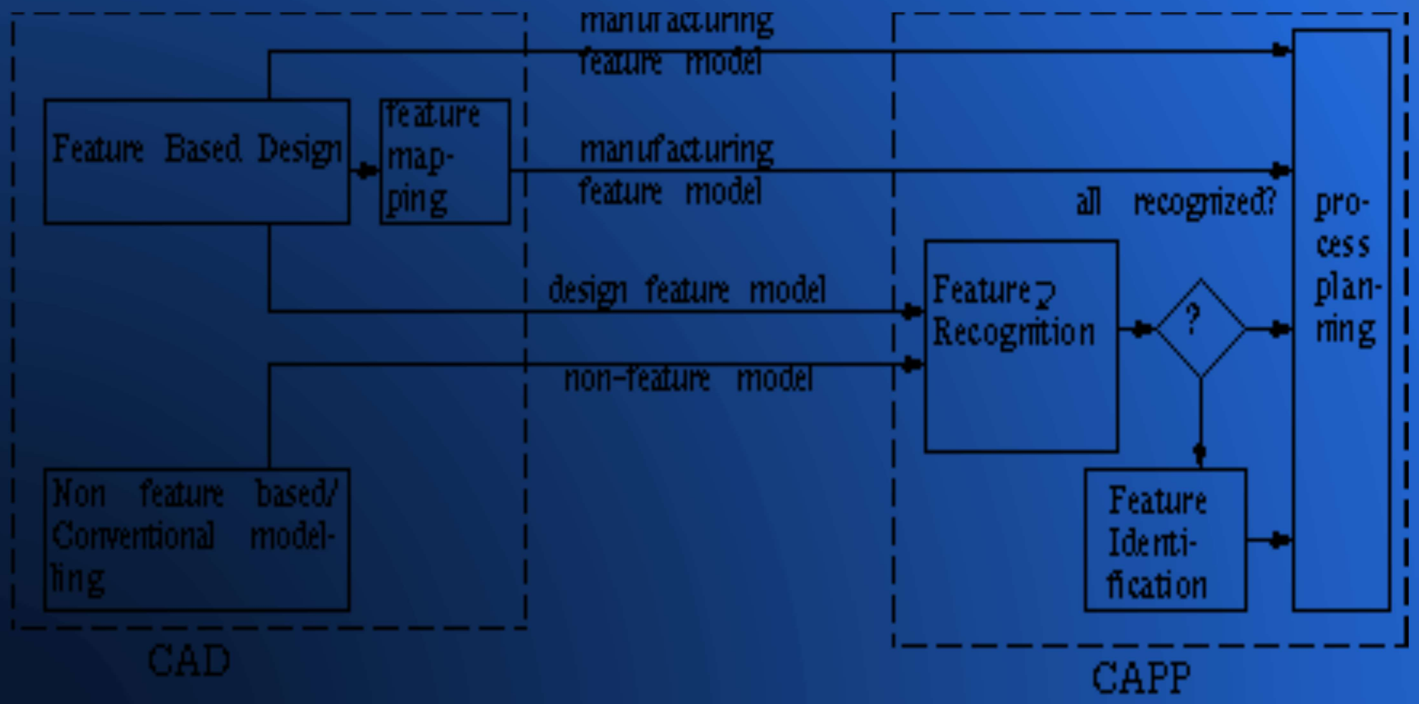


Figure 3.2 Ways of obtaining manufacturing features in a CAPP system.

Feature Based Modelling

- The word "Assembly" is normally used to refer to a 3D model that is composed of several distinguishable parts, that are put together in some way to create a functional object, just like real life products are made.
- For example, a bolt, a washer and a nut are three separate bodies that when put together comprise an assembly.
- Manual assembly In general, you don't need special tools to create assemblies, you just need to have many different bodies arranged in some way.
- To position the bodies where you want them, you can use the Std TransformManip tool;
- use the Std Placement.svg Std Placement dialog, or
- modify the placement property directly in the property editor.

Feature Based Modelling

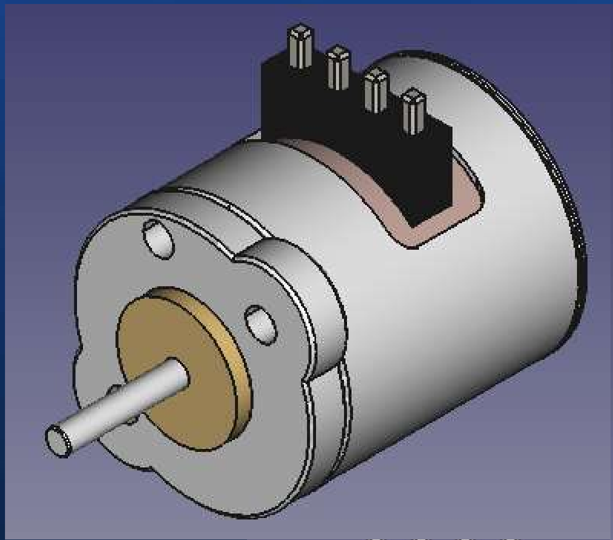
- You may use one of the pseudo-assembly external workbenches, like Lattice2, Manipulator, Part-o-magic, or WorkFeature, to help you find intersections, measure distances, and distribute your objects in the desired way.
- In general, the Std Part.svg Std Part object was designed to serve as the basic building block to create assemblies. This object is used to group several bodies and move them together as a unit, that is, as a sub-assembly. Then this sub-assembly can be placed next to, or used inside of other sub-assemblies in order to create the final assembly.
- Constrained assembly:
- You can also use a dedicated assembly workbench, like A2p workbench.svg A2plus, Assembly3 workbench icon.svg Assembly3, or Assembly4 workbench icon.svg Assembly4. Please note that Assembly2 is unmaintained, so it is not recommended for new models.

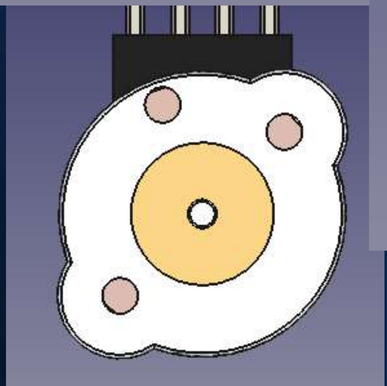
Feature Based Modelling

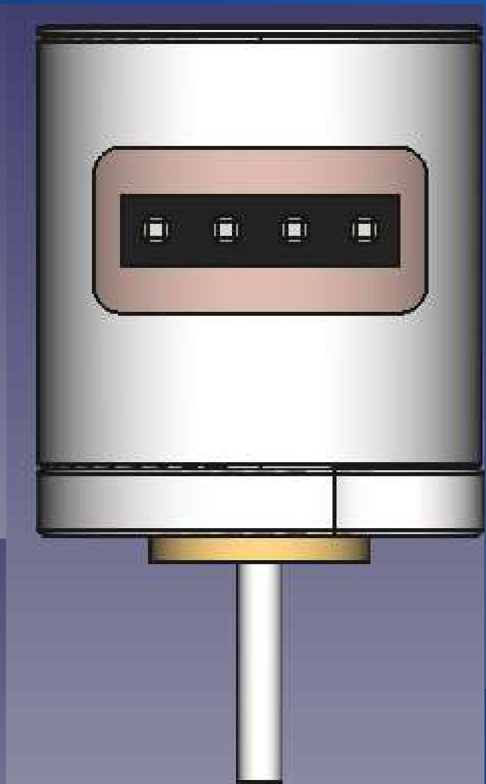
- The assembly workbenches use constraints and expressions to create relationships between the objects in your model, in order to mathematically tie the objects in place, for example, "this face should stick to this other face", "this cylinder should be concentric to that circle", "this point should follow this edge", etc.
- This is an advanced usage of the software that is normally used in complex mechanical systems. If your model is not very complex, then using an assembly workbench may not be necessary.
- https://wiki.freecad.org/A2plus_Workbench
- https://wiki.freecad.org/Assembly3_Workbench
- https://wiki.freecad.org/Assembly4_Workbench
- Please note that Assembly2 is unmaintained, so it is not recommended for new models.

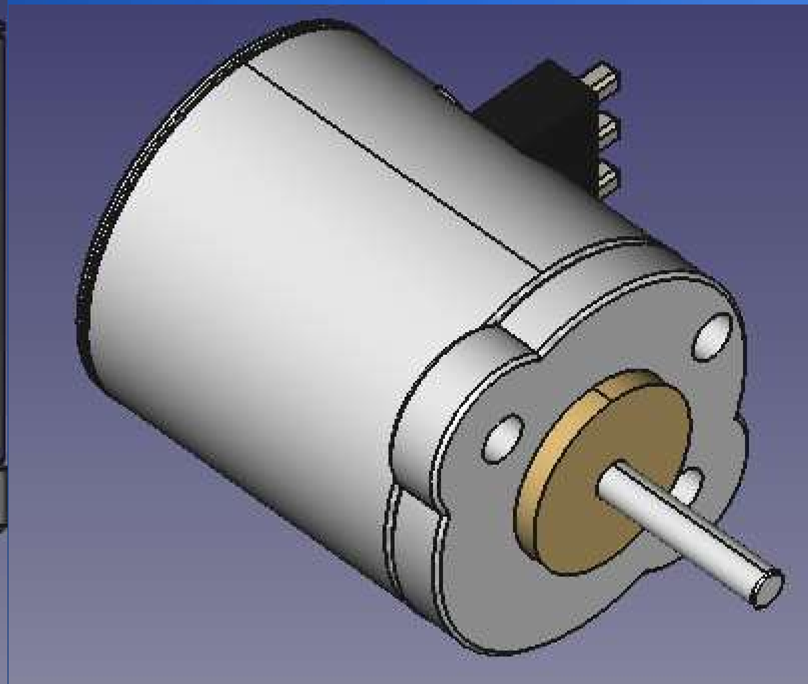
- For the FreeCAD commend reference, see:
- https://wiki.freecad.org/Category:Command_Reference For a complete list of commands.
-

3D Model of a stepper Motor

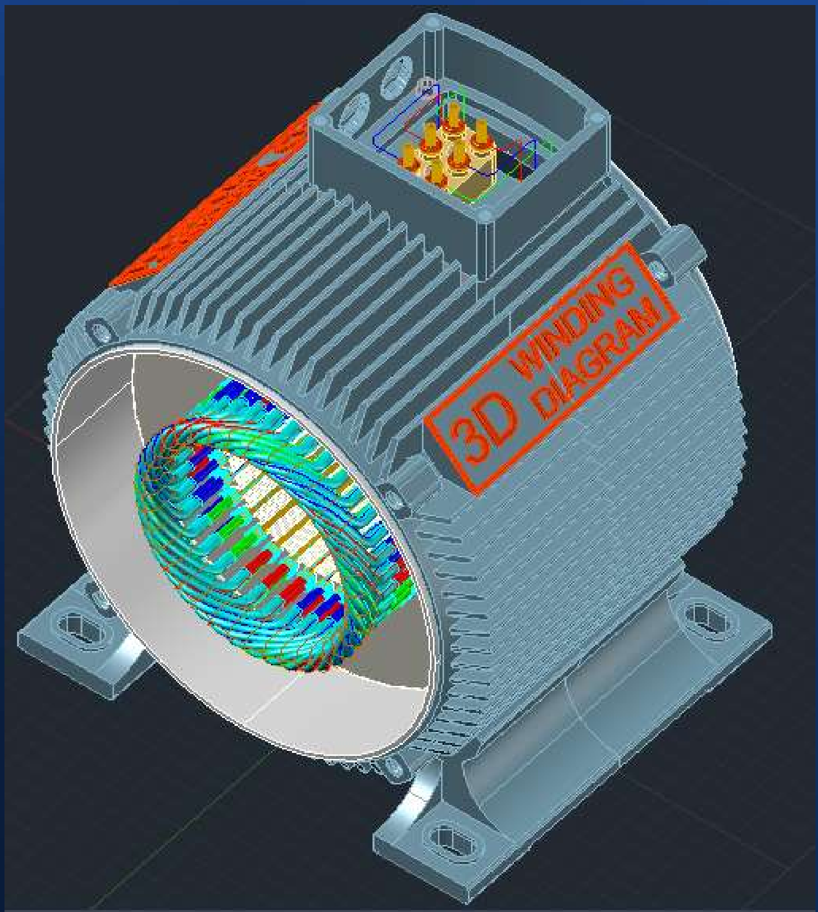


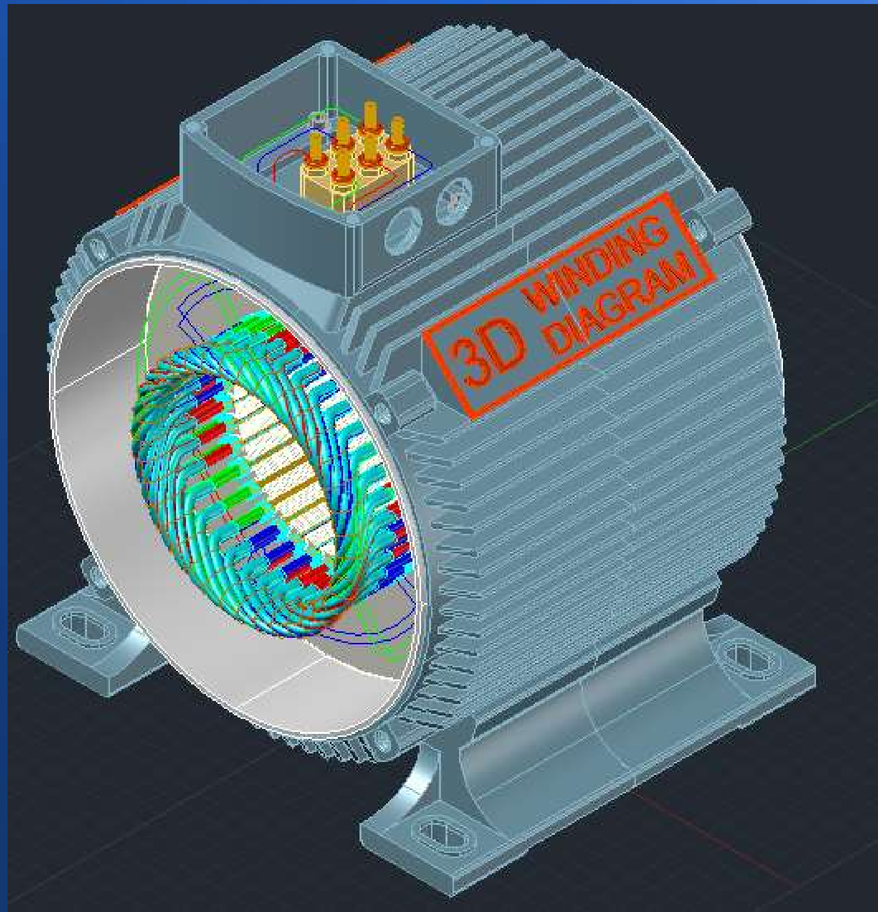






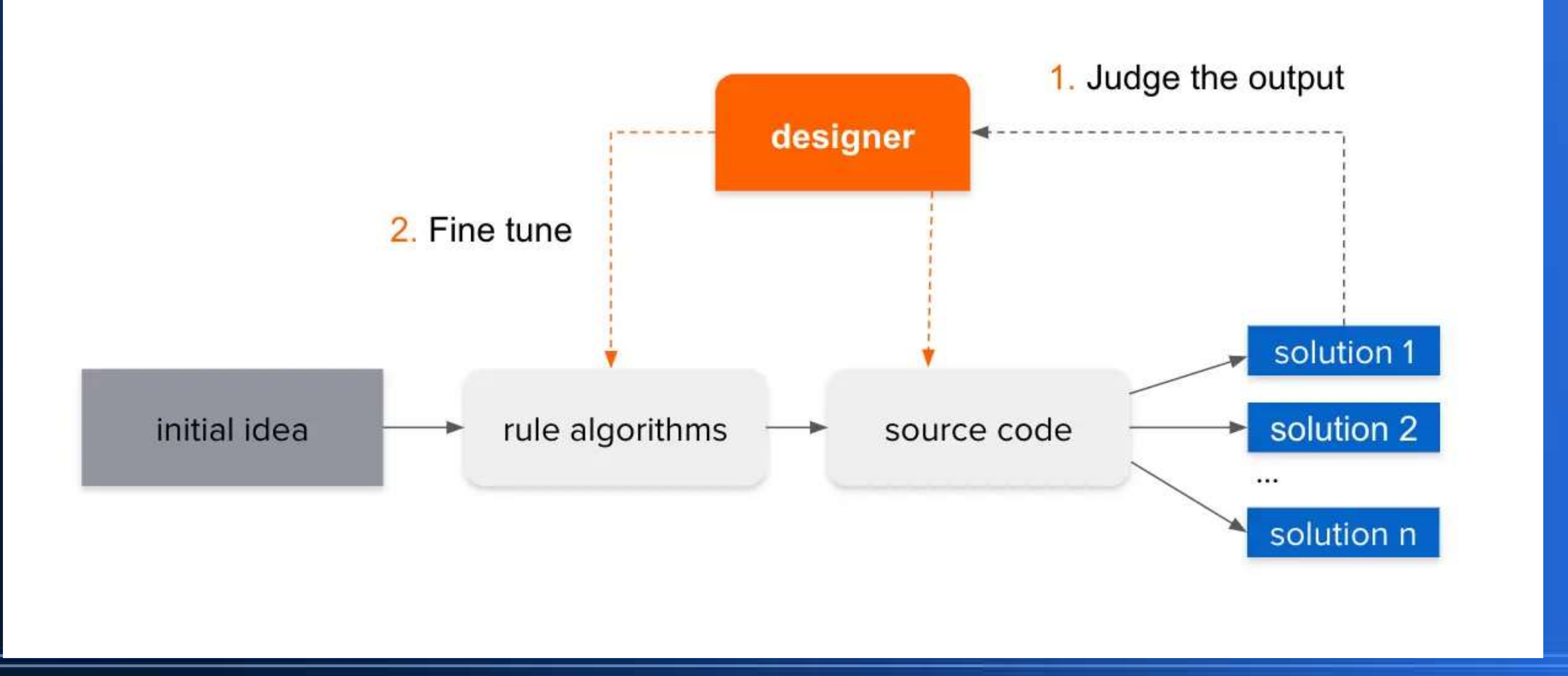
A 3 Phase motor in isometric view





Advanced Modeling: Surface modelling & Generative Modeling

- Generative design is an iterative design exploration process that uses an AI-driven software program to generate a range of design solutions that meet a set of constraints. Unlike traditional design, where the process begins with a model based on an engineer's knowledge, generative design begins with design parameters and uses AI to generate the model.



Advanced Modeling: Surface modelling & Generative Modeling

- By modifying the design parameters in an increasingly refined feedback loop, engineers can find highly optimised and customised design solutions to a wide range of engineering challenges, such as making product components lighter, stronger, and more cost-effective.
- The Difference Between Topology Optimisation and Generative Design:
- Generative design and topology optimisation have become buzzwords in the CAD design space, but it’s a common misconception that they’re synonymous.
- Topology optimisation isn’t new. It has been around for at least 20 years and has been widely available in common CAD software tools. The start of its process requires a human engineer to create a CAD model, applying loads and constraints with project parameters in mind. The software then generates a single optimised mesh-model concept ready for an engineer’s evaluation. In other words, topology optimisation requires a human-designed model from the outset to function, limiting the process, its outcomes, and its scale.
- In a way, topology optimisation serves as the foundation for generative design. Generative design takes the process a step further and eliminates the need for the initial human-designed model, taking on the role of the designer based on the predefined set of constraints.

- Surface Modeling
- Surface Modeling is the method of showing or presenting solid objects. The process requires you to convert between different 3D modeling types, such as converting the 3D object to show procedural surfaces, validate imperfections, and apply smoothness. While more complex than Wireframe Modeling, Surface Modeling is easier to achieve than Solid Modeling.
- In a CAD environment it is possible to fully produce a model using only surface modeling - but most real-world applications require a combination with solid modeling technique as well. How a designer is able to interact and manipulate an object in a CAD system is the difference between solid and surface modeling.

- Surface Modeling Explained
- Surface modeling gives you the ability to build out a visual representation of an object’s exterior and its contours. In other words, it’s a surface.
- These objects can be mechanical components like an engine - or complex organic shapes like animals.
- No matter what you’re designing, surface modeling requires you to define the exterior curves and shapes of your objects.
- Unlike solid modeling, your object can be geometrically and physically incorrect - with no properties of mass defined and no thickness. This gives the designer the unique ability to modify the model in ways that solid models are incapable.
- This also means, however, that surface models cannot be sliced open like their solid counterparts, because the object is essentially hollow.
- Also, note that these surfaces can be represented using NURBS or polygons, depending on the application.

- Surface Modeling Explained
- Surface modeling gives you the ability to build out a visual representation of an object's exterior and its contours. In other words, it's a surface.
- These objects can be mechanical components like an engine - or complex organic shapes like animals.
- No matter what you're designing, surface modeling requires you to define the exterior curves and shapes of your objects.
- Unlike solid modeling, your object can be geometrically and physically incorrect - with no properties of mass defined and no thickness. This gives the designer the unique ability to modify the model in ways that solid models are incapable.
- This also means, however, that surface models cannot be sliced open like their solid counterparts, because the object is essentially hollow.
- Also, note that these surfaces can be represented using NURBS or polygons, depending on the application.

Advanced Modeling: Surface modelling & Generative Modeling

- Surface modeling uses B-splines and Beiser equations in order to define the features of your solid appearing object - and is used in architectural illustrations and 3d animation for video games. This gives the following distinct advantages over other techniques:
- Making Changes To An Imported Model:
- When you import existing models into a CAD environment that it wasn’t originally built, it can present a host of issues - most notably it opens without any details of the features it’s made up of.
- Making changes to complex surfaces without these details can be extremely challenging. The flexibility of surface modeling allows you to delete or replace faces of the model in order to make your desired changes.
- Solid modeling tools have their limitations - like having to build out several sides of a shape at once. Surface modeling lets you build out faces individually, giving you control over the exact contour and direction.

Advanced Modeling: Surface modelling & Generative Modeling

- The Changing Role of the Engineer in Generative Design
- The role of the engineer has evolved alongside technology. As design has come to rely more heavily on computer software, understanding how to manipulate digital tools to solve design challenges has become a central requirement for engineers working in all industries. Before generative design, engineers would conceptualise and test iterations using hands-on sketching and modeling.
- Now, using generative design software, engineers specify high-level performance requirements and general design frameworks and then relegate the details to the software. The process of defining these parameters can be particularly complex when characterising novel materials or modeling hard-to-define problems and solution spaces.
- In a fundamental shift from traditional design processes, engineers will no longer need to create design solutions themselves. Instead, through generative design, they articulate and refine the environment in which design solutions can succeed. By letting computers do the “thinking,” generative design lets engineers focus on innovation and high-level problem-solving.

Advanced Modeling: Surface modelling & Generative Modeling

- Software Tools for Generative Design
- Individuals who have experience with CAD can easily make the leap to generative design software. In addition to generative design-specific software, many CAD programs now offer integrated generative design tools or plug-ins.
- Generative design software, however, offers users more than the traditional functionality of CAD software. These tools enable users to input information on forces, materials, costs, and the like into design profiles as well as prioritise and refine parameters based on graphical representations of design solutions.
- The following are popular software programs offering generative design capabilities:

Advanced Modeling: Surface modelling & Generative Modeling

- Fusion 360 from Autodesk: Fusion 360 offers users a powerful set of modeling tools, including sketching, direct modeling, surface modeling, parametric modeling, mesh modeling, rendering, and much more. Its generative design capabilities enable users to identify design requirements, constraints, materials, and manufacturing options to generate manufacturing-ready designs, all the while enabling users to leverage the power of machine learning and AI to review cloud-generated design outcomes based on visual similarities, plots, and filters.
- Creo Generative Design from PTC: Leveraging the cloud, this software enables users to create optimised design concepts and simultaneously explore and test numerous design iterations quickly. It highlights the iterations that best match a user’s objectives based on design parameters the user sets. Within the Creo design environment, this software promises to generate high-quality, lower-cost, and manufacturable designs, all in less time than leading competitors.

Advanced Modeling: Surface modelling & Generative Modeling

- nTop Platform from nTopology: The nTop Platform software promises users complete control over every aspect of the optimisation process and its outputs. Leveraging advanced generative tools, users can create custom, reusable workflows tailored to an application's unique requirements. Featured capabilities of this program include unbreakable modeling and latticing operations, topology optimisation, reusable design workflows, field-driven design, and mechanical-thermal finite element analysis simulations.
- NX from Siemens: Beyond generative design, the main feature that NX offers is the digital twin technology, which promises users a flexible, powerful, and integrated solution to help them streamline the design and delivery of better products. NX combines design interoperability, validation, model-based definition, and more to help users move products through research and development faster and at lower costs while improving product quality.
- MSC Apex Generative Design from MSC Software: This program promises users an end-to-end solution for making high-precision metal components more quickly and with less human intervention than its competitors. MSC Software reports that users experience reductions in initial design and setup time by as much as 80 percent. At a glance, the software combines simplicity, automated design, import and validation, and direct output in one process.

Advanced Modeling: Surface modelling & Generative Modeling

- Applications of Generative Design
- Generative design applications exist across many industries—from aerospace and architecture to manufacturing and consumer goods. Engineers who use generative design are often trying to solve complex engineering challenges. Such challenges include reducing component weights and manufacturing costs, scaling component customisation, and optimising performance.
- For example, in the automotive manufacturing industry, engineers utilise generative design to reduce component weights, improve weak design areas, decrease production costs through component consolidation, and reduce the time to market for new products.
- Similarly, in the sports equipment industry, designers leverage generative design to achieve new levels of product performance while minimising production costs. In the aerospace industry, generative design enables airline manufacturers to reduce the weight and improve the strength of plane components, helping airlines reduce fuel consumption to lower costs and emissions as a result.

Advanced Modeling: Surface modelling & Generative Modeling

- Photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring, and interpreting photographic images and patterns of recorded radiant electromagnetic energy and other phenomena.
- Photogrammetry is nearly as old as photography itself. Since its development approximately 150 years ago, photogrammetry has moved from a purely analog, optical–mechanical technique to analytical methods based on computer-aided solution of mathematical algorithms and finally to digital or softcopy photogrammetry based on digital imagery and computer vision, which is devoid of any opto-mechanical hardware. Photogrammetry is primarily concerned with making precise measurements of three-dimensional objects and terrain features from two-dimensional photographs. Similar techniques can be applied in CAD.

Advanced Modeling: Surface modelling & Generative Modeling

- Generative Design and 3D Printing: Why Generative Design and 3D Printing Work Well Together:
- Generative design algorithms often create highly efficient organic shapes with supporting latticework that are costly or even impossible to fabricate with conventional manufacturing technologies such as injection molding or subtractive manufacturing tools like CNC machining. Advanced manufacturing tools like 3D printing—also known as additive manufacturing—are essential for the types of performance-sensitive applications where this optimisation is compelling.
- 3D printing works well with generative design because it provides a flexible, fast means of producing a three-dimensional, high-resolution model of one or more design iterations for a cost-effective final product. Generally, in economic terms, the cost-effectiveness of 3D printing increases alongside the complexity of a design iteration.
- 3D printing is more cost-competitive at lower production volumes because you do not need to reach an economy of scale to offset setup costs. Therefore, it facilitates the mass customisation that generative design makes possible. As the cost of 3D printing continues to decrease and the variety of materials increase, 3D printing is becoming practical for small and mid-volume parts for more and more applications.

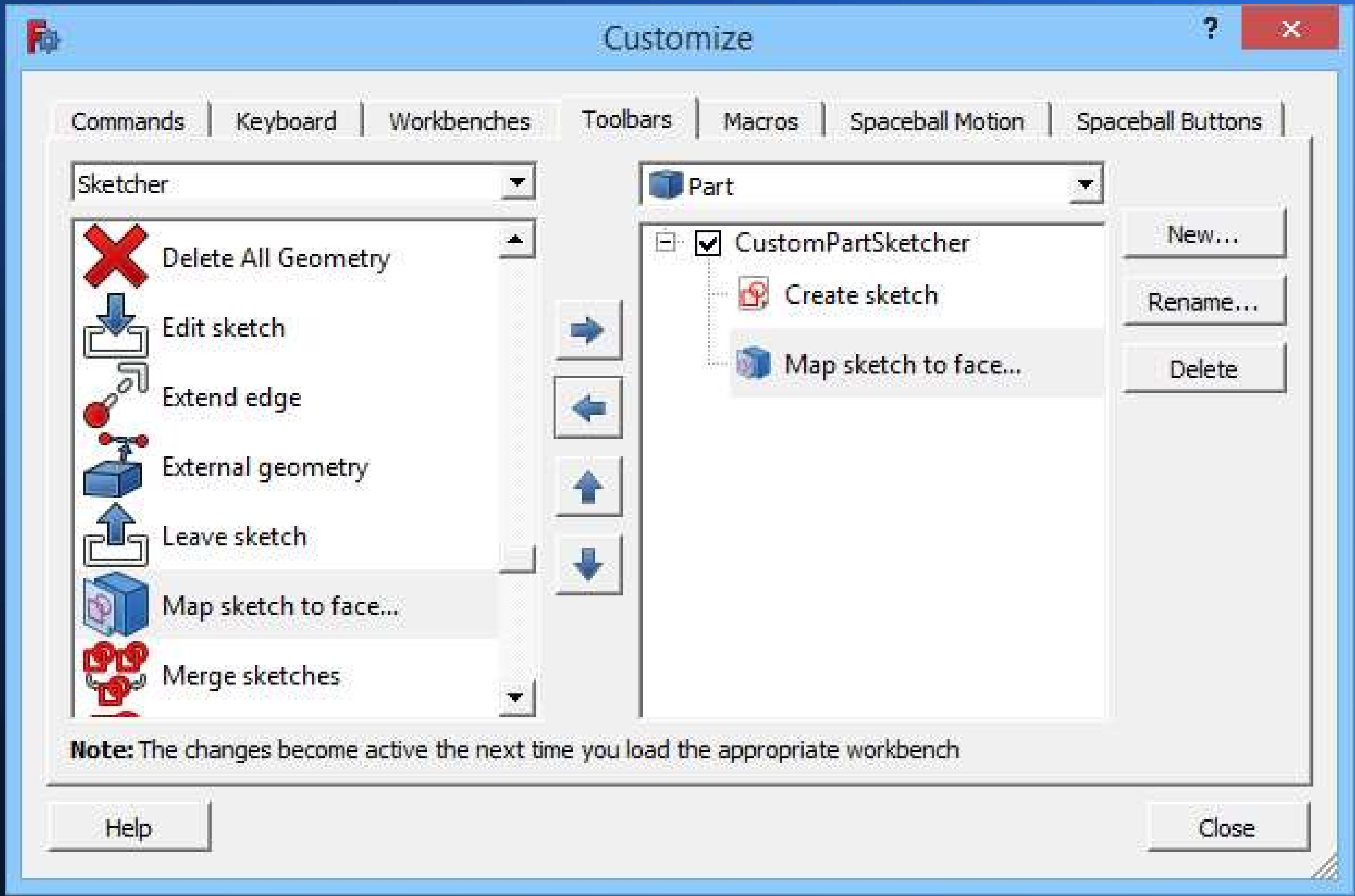
Advanced Modeling: Surface modelling & Generative Modeling

- Innovating the Future
- Generative design plays an increasingly central role in the design of products across a wide range of industries. Whether a company aims to reduce the weight of an aircraft engine bracket, make an electric wheelchair more portable, or customise a running shoe, generative design and 3D printing are paving the way for a fully optimised and customised future.
- As advancements in AI and additive manufacturing continue to extend the horison of possibility for generative design, more applications and benefits of these innovative and cutting-edge technologies will emerge.

Customising of CAD systems using macros and Application Programming Interface (API)

- The FreeCAD interface is based on the modern Qt toolkit and has a state-of-the-art organisation. Some aspects of the interface can be customised. You can, for example, add custom toolbars, with tools from several workbenches or tools defined in macros, and you can create you own keyboard shortcuts. But the menus and default toolbars that come with FreeCAD and its workbenches cannot be changed.
- The commands available in the Customise dialog box depend on the workbenches that have been loaded in the current FreeCAD session. So you should first load all workbenches whose commands you want to have access to. There are several ways to invoke the Std DlgCustomise.svg Std DlgCustomise command:Select the Tools → Std DlgCustomise.svg Customise... option from the menu.Right-click a toolbar area and choose Std DlgCustomise.svg Customise... from the context menu.The Customise dialog box opens. For more information see Options.

Customising of CAD systems using macros and Application Programming Interface (API)



Advanced Modeling: Surface modelling & Generative Modeling

- PLaSM (Programming Language of Solid Modeling) is an open source scripting language for solid modeling, a discipline that constitutes the foundation of computer-aided design and CAD systems. In contrast to other CAD programs, PLaSM emphasizes scripting rather than interactive GUI work. Users can create arbitrarily complex designs using a wide range of simple 2D and 3D objects, advanced curves and curved surfaces, Boolean operations, and elementary as well as advanced geometric transformations.
- The scripting approach is very different from working with an interactive Graphical User Interface (GUI). Although it means less user comfort, it is preferred by numerous CAD instructors[which?] as scripts reveal all details of the design procedure (not only the final design) and students are exposed to elementary computer programming.
- PLaSM has been developed since the 1980s by the CAD group at the Universities Roma Tre and La Sapienza in Rome, Italy, by Alberto Paoluzzi and his collaborators. It was used at the University of Rome to create an extensive database of ancient Rome architecture.

Constructive solid geometry

- Constructive solid geometry (CSG; formerly called computational binary solid geometry) is a technique used in solid modeling. Constructive solid geometry allows a modeler to create a complex surface or object by using Boolean operators to combine simpler objects, potentially generating visually complex objects by combining a few primitive ones.
- In 3D computer graphics and CAD, CSG is often used in procedural modeling. CSG can also be performed on polygonal meshes, and may or may not be procedural and/or parametric.
- Contrast CSG with polygon mesh modeling and box modeling.
- The simplest solid objects used for the representation are called geometric primitives. Typically they are the objects of simple shape: cuboids, cylinders, prisms, pyramids, spheres, cones.[1] The set of allowable primitives is limited by each software package. Some software packages allow CSG on curved objects while other packages do not.
-

Advanced Modeling: Surface modelling & Generative Modeling

- An object is constructed from primitives by means of allowable operations, which are typically Boolean operations on sets: union, intersection and difference, as well as geometric transformations of those sets.
- A primitive can typically be described by a procedure which accepts some number of parameters; for example, a sphere may be described by the coordinates of its center point, along with a radius value. These primitives can be combined into compound objects using operations like these:
- Union: Merger of two objects into one
- Difference: Subtraction of one object from another
- Intersection: Portion common to both objects
- Combining these elementary operations, it is possible to build up objects with high complexity starting from simple ones.

Advanced Modeling: Surface modelling & Generative Modeling

- Ray tracing
- Rendering of constructive solid geometry is particularly simple when ray tracing. Ray tracers intersect a ray with both primitives that are being operated on, apply the operator to the intersection intervals along the 1D ray, and then take the point closest to the camera along the ray as being the result.
- Constructive solid geometry has a number of practical uses. It is used in cases where simple geometric objects are desired,[citation needed] or where mathematical accuracy is important. Nearly all engineering CAD packages use CSG (where it may be useful for representing tool cuts, and features where parts must fit together).

Customising of CAD systems using macros and Application Programming Interface (API)

- Automating the design process of a product or a system can provide engineers and designers with many benefits. As such, repeatable tasks that are time consuming can be handled automatically and with minimal human attention. This is achieved by using the API (Application Programmable Interface) of CAD systems in order to create macros or even develop software applications.This application automates the design process.
- The design process begins with the creation of a series of components developed as solids, and extends to the extraction of basic attributes and properties from the complete mechanical assembly.
- As we are using FreeCAD, I will discuss the process of using macros and automation in FreeCad.
- For a demonstration see: http://dfdn.info/video/FreeCAD-Sketching-Demonstration/FreeCAD-Sketching-Demonstration.html

Customising of CAD systems using macros and Application Programming Interface (API)

- Python is a popular, open source programming language, very often embedded in applications as a scripting language, as is the case with FreeCAD. It has a series of features that make it suitable for us FreeCAD users: It is very easy to learn, especially for people who had never programmed before, and it is embedded in many other applications. This make it a valuable tool to learn, as you will be able to use it in other software, such as Blender, Inkscape or GRASS.
- FreeCAD makes extensive use of Python. With it, you can access and control almost any feature of FreeCAD. For example, you can create new objects, modify their geometry, analyze their contents, or even create new interface controls, tools and panels. Some FreeCAD workbenches and most of the addon workbenches are fully programmed in Python. FreeCAD has an advanced Python console, available from menu View->Panels->Python console. It is often useful to perform operations for which there is no toolbar button yet, or to check shapes for problems, or to perform repetitive tasks:

Customising of CAD systems using macros and Application Programming Interface (API)

- But the Python console has another very important use: Every time you press a toolbar button, or perform other operations in FreeCAD, some Python code is printed in the console (if the option to Show script commands in Python console is enabled in Edit → Preferences → General → Macro) and executed. By leaving the Python console open, you can literally see the Python code unfold as you work, and in no time, almost without knowing it, you will find yourself learning some of the Python language.
- FreeCAD also has a macros system, which allows you to record actions to be replayed later. This system also uses the Python console, by simply recording everything that is done in it.

Customising of CAD systems using macros and Application Programming Interface (API)

- Writing Python code
- There are two easy ways to write Python code in FreeCAD: From the Python console (View → Panels → Python Console), or from the Macro editor (Tools → Macros → New). In the console, you write Python commands one by one, which are executed when you press return, while macros can contain a more complex script made of several lines, which is executed only when the macro is launched from the same Macros window.
- In this example, you will be able to use both methods, but it is highly recommended to use the Python Console, since it will immediately inform you of any errors you make while typing. If you are interested in learning more, the FreeCAD documentation has an extensive section related to Python programming:
- https://wiki.freecad.org/Power_users_hub

Customising of CAD systems using macros and Application Programming Interface (API)

- Writing Python code
- There are two easy ways to write Python code in FreeCAD: From the Python console (View → Panels → Python Console), or from the Macro editor (Tools → Macros → New). In the console, you write Python commands one by one, which are executed when you press return, while macros can contain a more complex script made of several lines, which is executed only when the macro is launched from the same Macros window.
- In this section, you will be able to use both methods, but it is highly recommended to use the Python Console, since it will immediately inform you of any errors you make while typing.
- If this is your first time using Python, consider reading this short introduction to Python programming before going any further, it will make the basic concepts of Python clearer.
- https://wiki.freecad.org/Introduction_to_Python

Python scripting

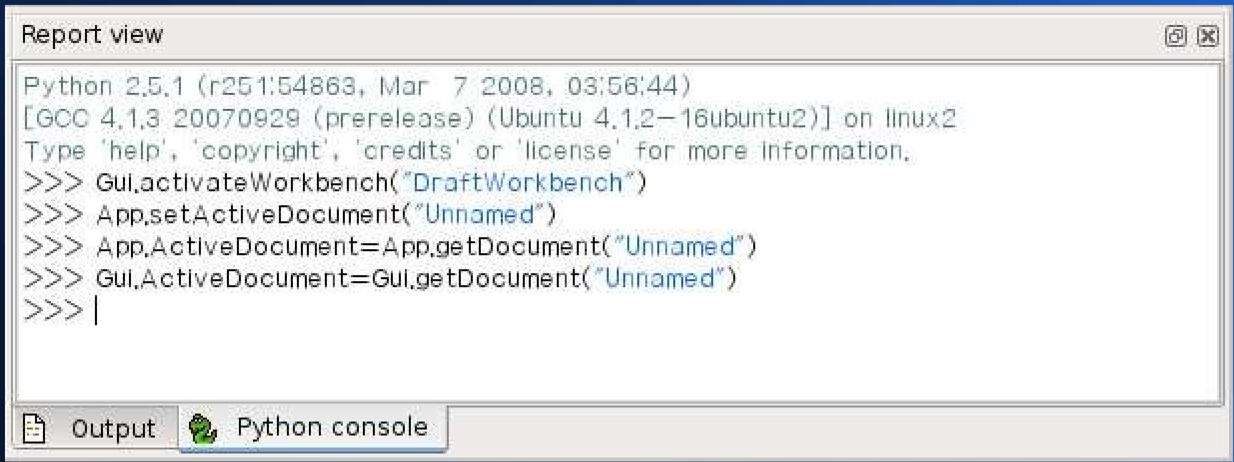
- Python is a programming language that it relatively easy to learn and understand. It is open-source and multi-platform, and can be used for many purposes: from simple shell scripts to very complex programs. But its most widespread use is as a scripting language embedded in other applications. That is how it is used inside FreeCAD. From the Python console, or from custom scripts, you can control FreeCAD and make it perform very complex operations.
- For example, from a Python script, you can:
- Create new objects.
- Modify existing objects.
- Modify the 3D representation of those objects.
- Modify the FreeCAD interface.
- There are several ways to use Python in FreeCAD:

Python scripting

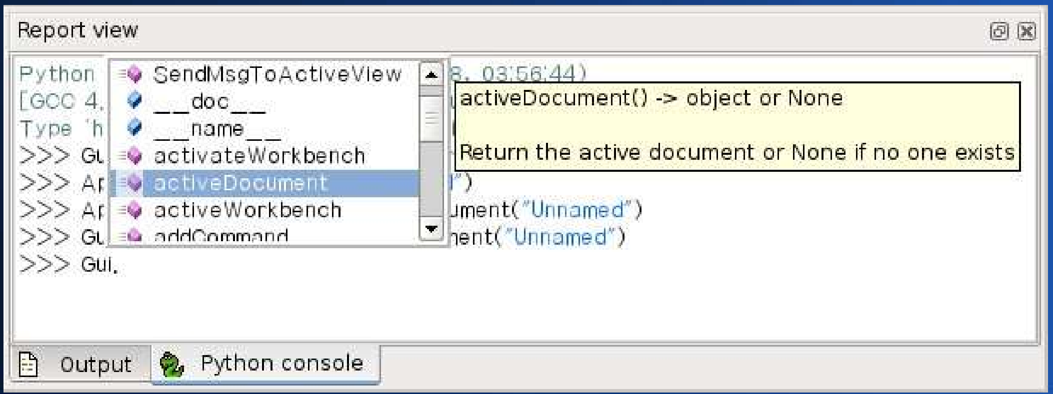
- From the FreeCAD Python interpreter, where you can issue commands in a "command line"-style interface.
- From macros, which are a convenient way to quickly add a missing tool to the FreeCAD interface.
- From external scripts, which can used to create quite complex solutions, even entire Workbenches.
- In this tutorial, we'll work on a couple of basic examples to get you started, but there is much more documentation about Python scripting available on this wiki. If you are totally new to Python and want to understand how it works, we also have a basic introduction to Python.
- Before proceeding with Python scripting, go to Edit → Preferences → General → Output window and check two boxes:
- Redirect internal Python output to report view.
- Redirect internal Python errors to report view.

Python scripting

- Then go to View → Panels and check: Report view.
- Writing Python code:
- There are two ways to write Python code in FreeCAD. In the Python console (select View → Panels → Python console from the menu) or in the Macro editor (select Macro → Macros... from the menu). In the console you write Python commands one by one, executing them by pressing Enter, while macros can contain more complex code made up of several lines, executed only when the macro is executed.



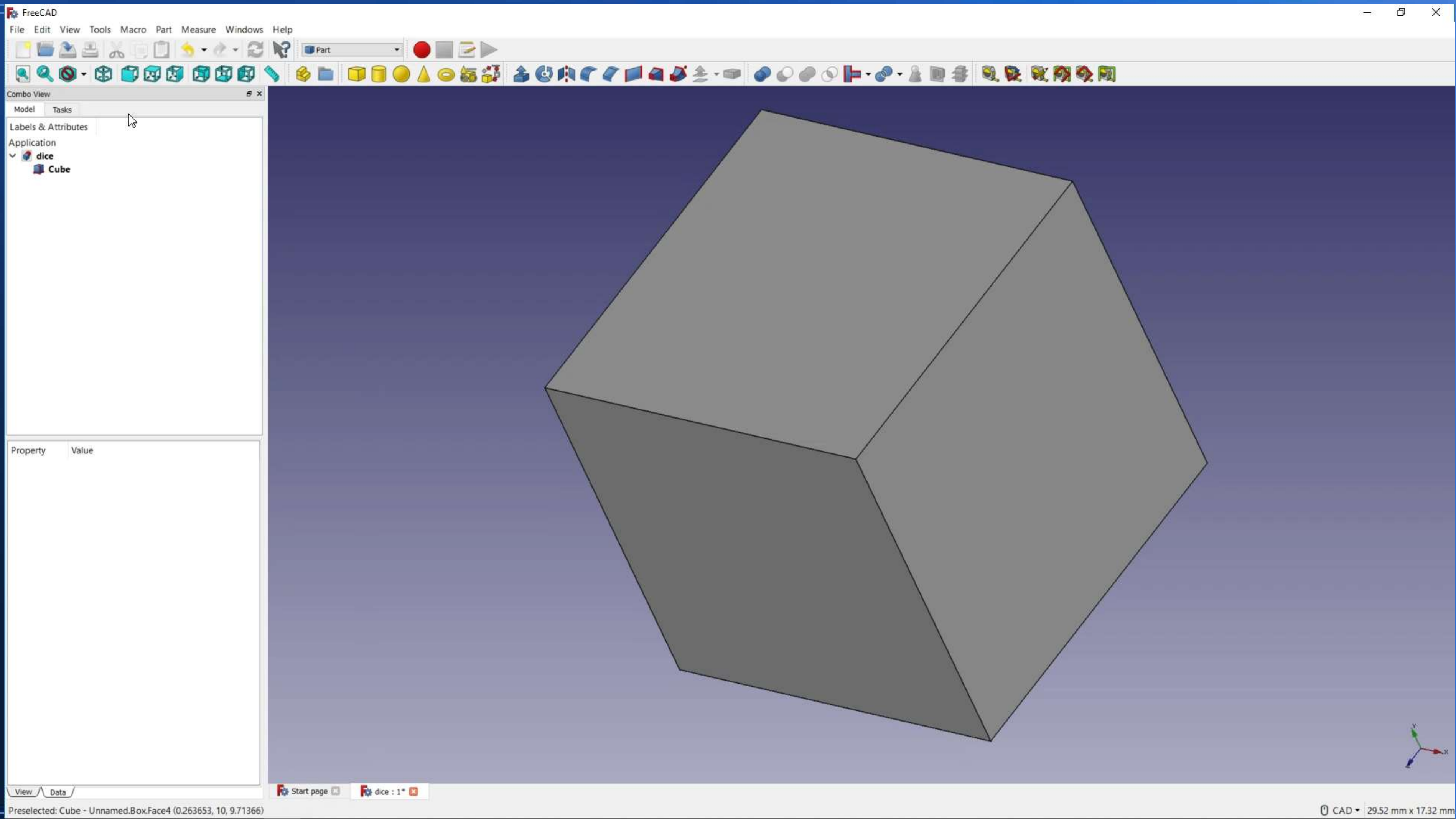
Python scripting

- Exploring FreeCAD:
- Let's start by creating a new empty document:
- doc = FreeCAD.newDocument()
- If you type this in the FreeCAD Python console, you will notice that as soon as you type FreeCAD. a window pops up, allowing to quickly autocomplete the rest of your line. Even better, each entry in the autocomplete list has a tooltip explaining what it does. This makes it easier to explore the available functionality. Before choosing new Document, have a look at the other options.
- 

Project – Make a Dice

- Click on the Create a new empty document icon to get started.
- document-new
- Click on the Save the active document icon. Name the file you save Dice.
- document-save
- FreeCAD has many different Workbenches, each with its own set of tools.
- To begin, choose the Part workbench.
- In the Part workbench, you have access to primitive objects such as a cube, a cylinder, and a sphere. These are 3D objects that you can add to your project and then edit.
- Primitives:
- Click on the Cube icon to add a cube to your project.

Project – Make a Dice



Project – Make a Dice

- To manually alter the view, use:
- Ctrl + right mouse button to move the view
- Shift + right mouse button to rotate the view
- Ctrl + Shift + right mouse button to zoom the view
- The cube is currently 10mm × 10mm × 10mm, so it’s a bit too small. The cube is also not in the centre of the project. Your next task is to edit the cube’s parameters.
- Click on the Draw Style icon.
- DrawStyleAsIs
- Then click on the Wireframe mode icon in the left-hand menu.
- DrawStyleWireFrame
- Click on View and then Toggle axis cross to see the project’s central axis.

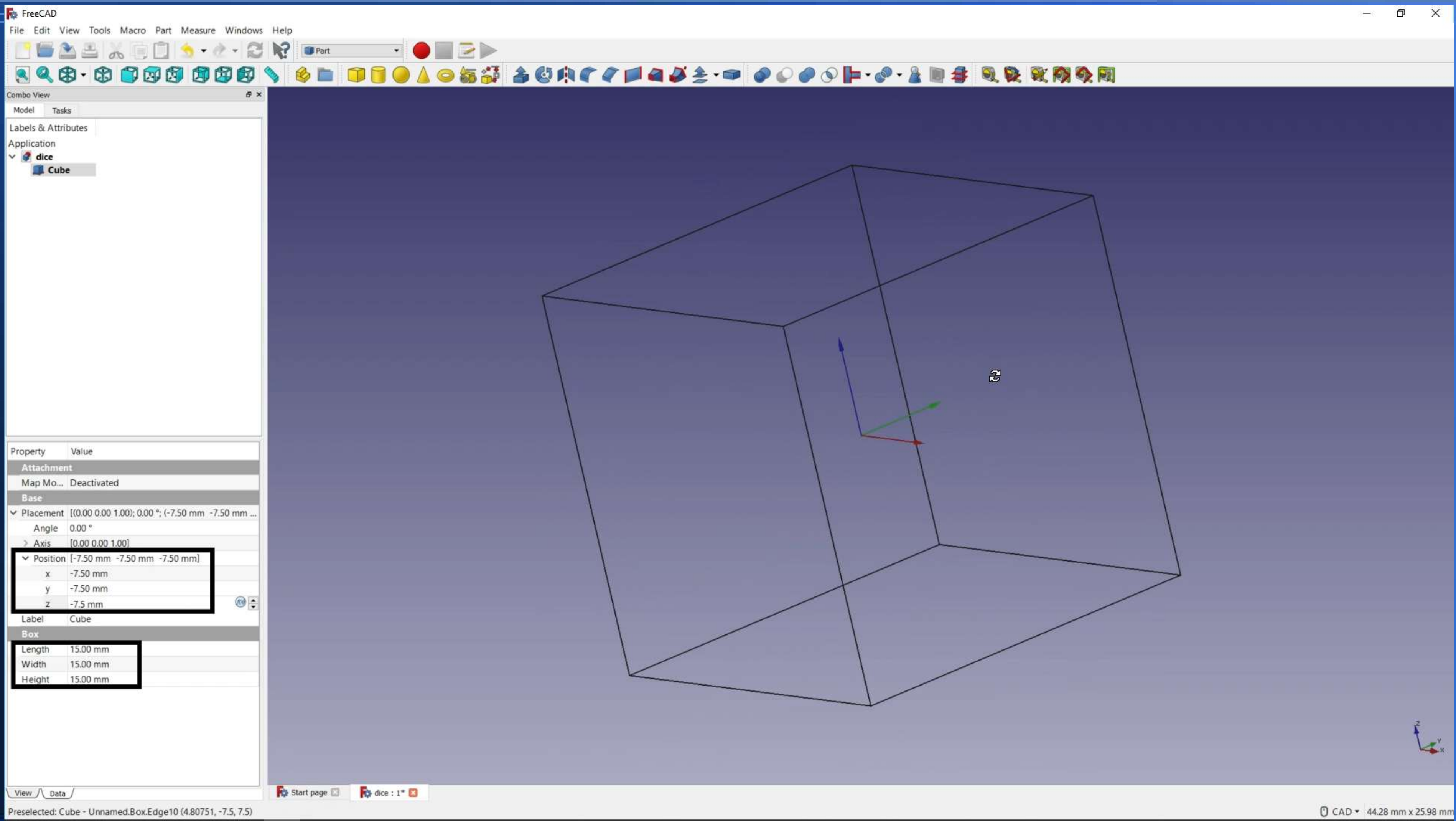
Project – Make a Dice

- With the cube selected in the Labels & Attributes view, make sure you have selected the Data tab at the bottom of the screen.
- Edit the cube's Length, Width, and Height.
- Then edit the cube's x, y, and z position of the cube.
- Length: 15mm
- Width: 15mm
- Height: 15mm
- Position:
- x: -7.5mm
- y: -7.5mm
- z: -7.5mm

PytProject – Make a Dicehon scripting

- edit-parameters_highlights
- You should see that the axis cross is now in the centre of the cube.
- Add construction geometry...
- Now that you have the basuc shape of the dice, prepare to add the dots that represent the numbers.
- To do this, you will sketch shapes onto the surface of the dice and then create pockets in the surface.
- First, you need to change to a different workbench.
- Switch to the Part Design workbench.
- Part-design →
- Make sure that your Common object is selected in the Model tab.

Project – Make a Dice



Project – Make a Dice

- Then click on the Create a new body and make it active icon.
- PartDesign_Body_Create_New
- When you click on the Model tab, your Common object should become grayed out.
- create-new-body
- Next, select the face of the dice to which you plan to add the first set of pips.
- Then click on the Create a new sketch icon.
- Sketcher_New_Sketch
- Your perspective on your dice should change, and the software should switch to the Tasks tab.
- task_tab

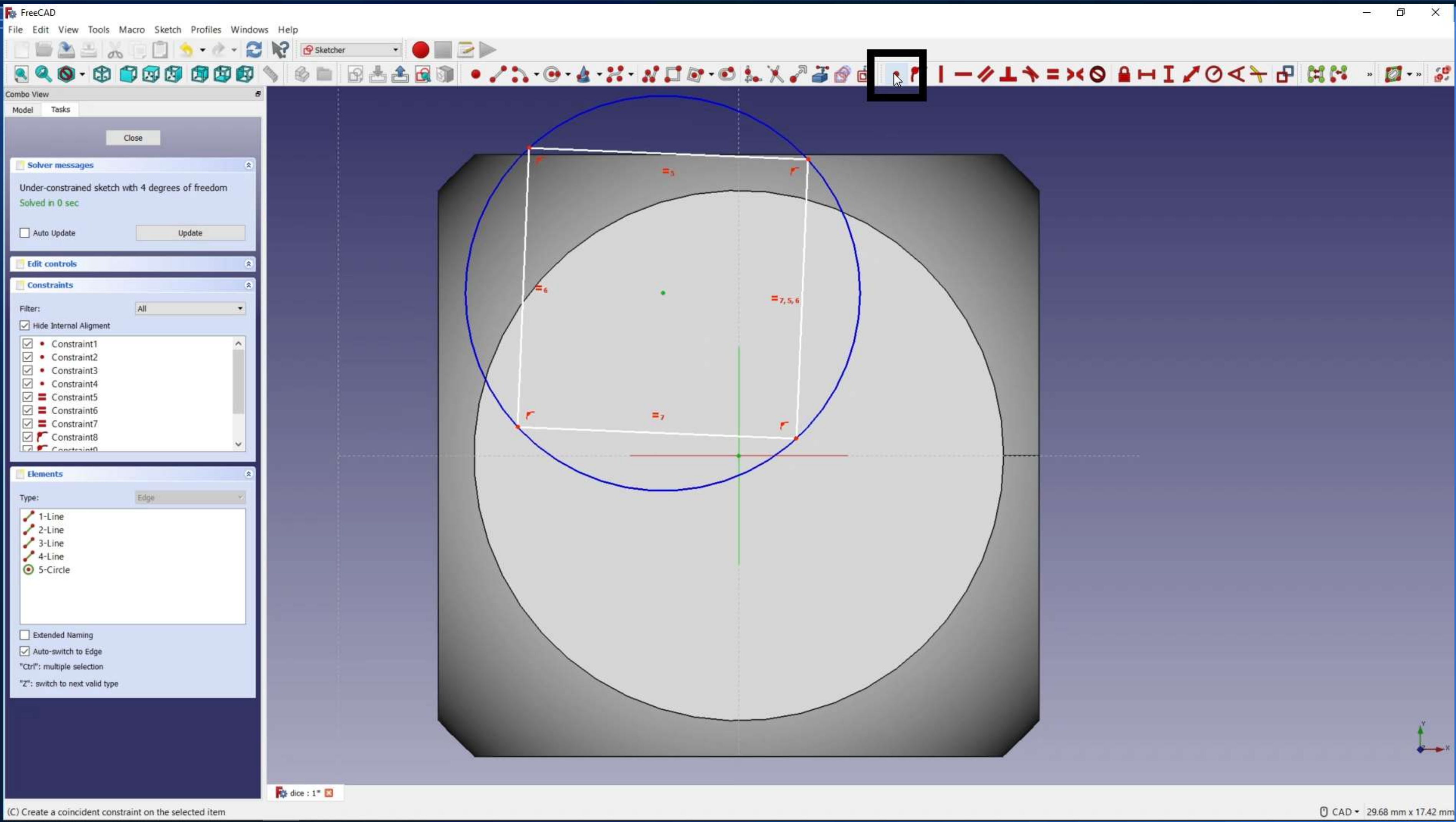
Project – Make a Dice

- To make sure that the pips are all placed accurately, you need to create a construction geometry next. A construction geometry is like a guide that helps you to accurately place the remaining sketch.
- Click on the drop-down arrow next to the Create a regular polygon in the sketcher icon.
- Sketcher_CreateHexagon
- Then click on the *Create a square by its center and by one corner icon.
- here are two ways to place the square:
- Either place your cursor over the central dot you can see in the cube, and then click and drag outwards
- Or draw the square anywhere, and then use a constraint to place it in the centre of the dice
- Place the square in the centre of the dice:

Project – Make a Dice

- Either carefully click on the central dot and then drag your cursor outwards to draw your square. The square can be of any size. Once you have your square, press the Escape key so you don't draw any more squares.
- In the Tasks tab, there is a message saying that you have an "Under-constrained sketch with 2 degrees of freedom".
- If you click on your square and drag an edge around a little, you see that you can both rotate and resize it. Rotation and size are the two degrees of freedom.
- You should always make sure that your sketches are fully constrained. This is how you do that for this sketch.
- Press the Esc key to deselect the current tool you are using.
- Click on one of the edges of your square. Then click on the horizontal or vertical axis (in red or green). Then click on the Create a parallel constraint between two lines icon to set these two lines to be parallel to each other.

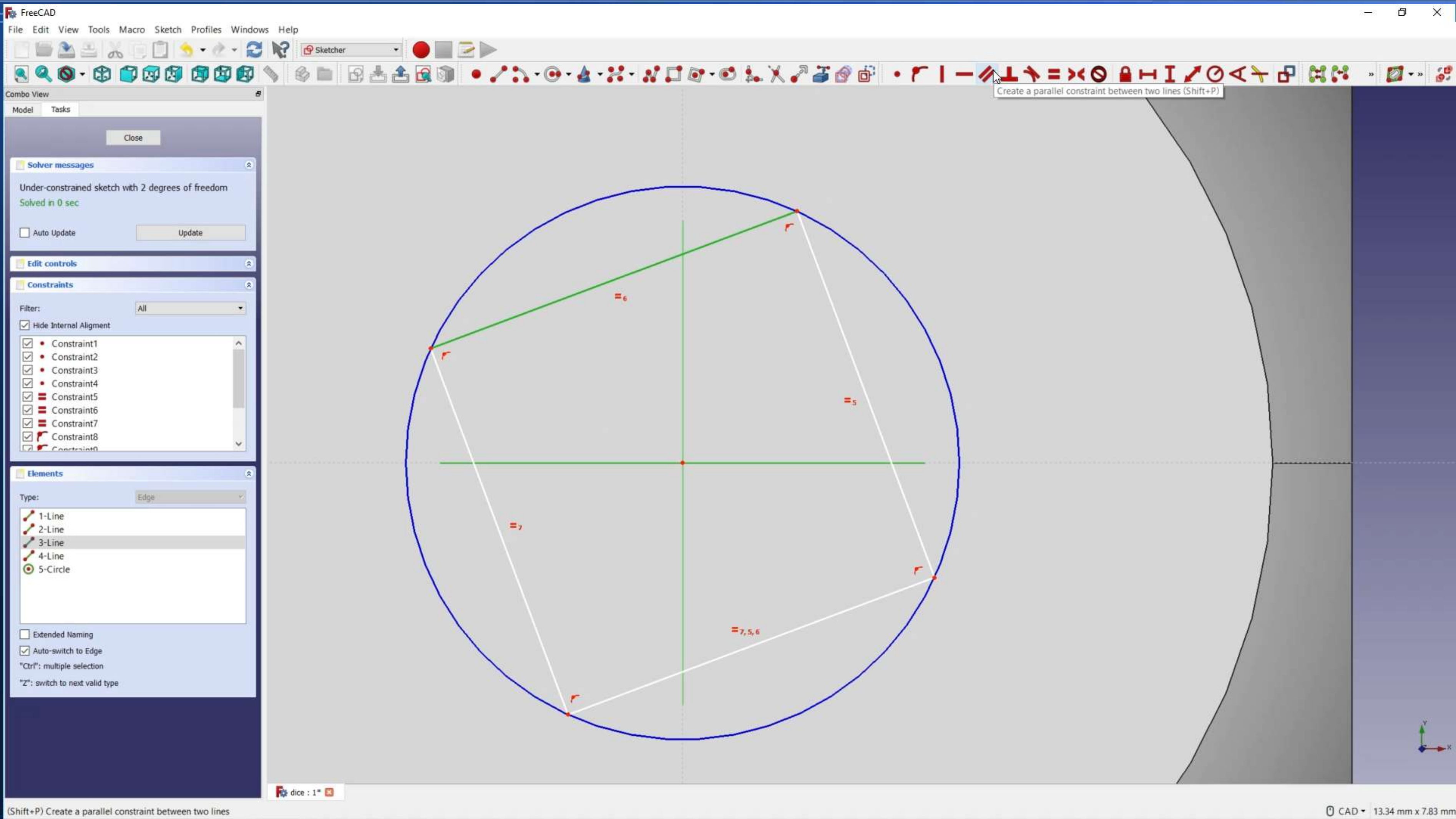
Project – Make a Dice



Project – Make a Dice

- This stops the square from being rotatable and reduces the number of degrees of freedom to 1.
- Click on the right edge of the square to select it. Then click on the Fix the vertical distance between two points or line ends icon and set the length of the line to 7mm.
- Constraint_VerticalDistance
- In the Tasks tab, you should now see the message “Fully constrained sketch”.
- Now change the square to a construction geometry.
- Hold down the Ctrl key and select all four lines of the square. Then click on the Toggles the toolbar or selected geometry to/from construction mode icon.
- Sketcher_AlterConstruction
- The square should turn blue.

Project – Make a Dice



Project – Make a Dice

- Draw dots
- Now that you have some construction geometry, you can add the dots to the surface of the dice.
- Click on the Create a circle in the Sketcher icon, and then draw six circles anywhere on the surface.
- Sketcher_CreateCircle
- six_circles.png
- In the Tasks tab, there is a new message saying that the sketch now has a large number of degrees of freedom.
- Click on the centre of a circle. Then hold down the Ctrl key and click on a corner of the construction geometry. Click on the Create a coincident constraint on the selected item icon to place the circle at that corner of the square.
- Repeat this with three more circles so there is a circle at each corner.

Project – Make a Dice

Then you vertically align

the remaining two circles with the

corners of the construction geometry.

Now align the two middle circles with each

other and with the centre of the

construction geometry.

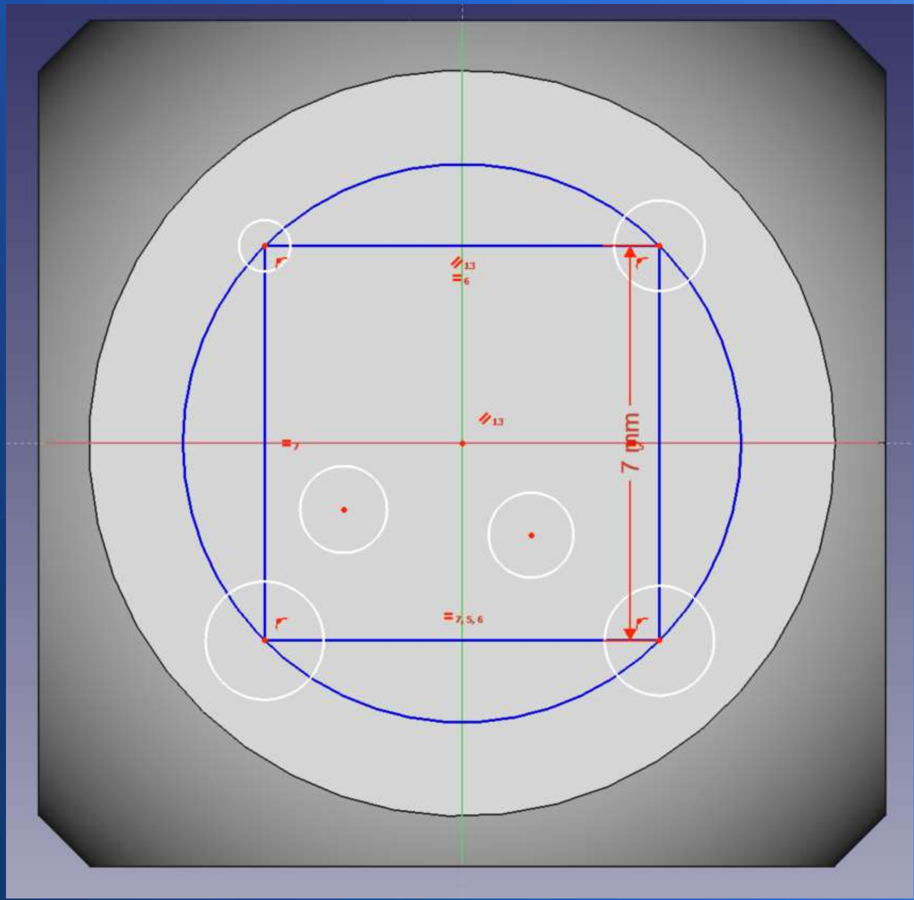
Click on the centre of the square. Then

hold down the Ctrl key and click on the

two middle circles. Click on the Create a

horizontal constraint on the selected item

icon to horizontally align all three shapes.



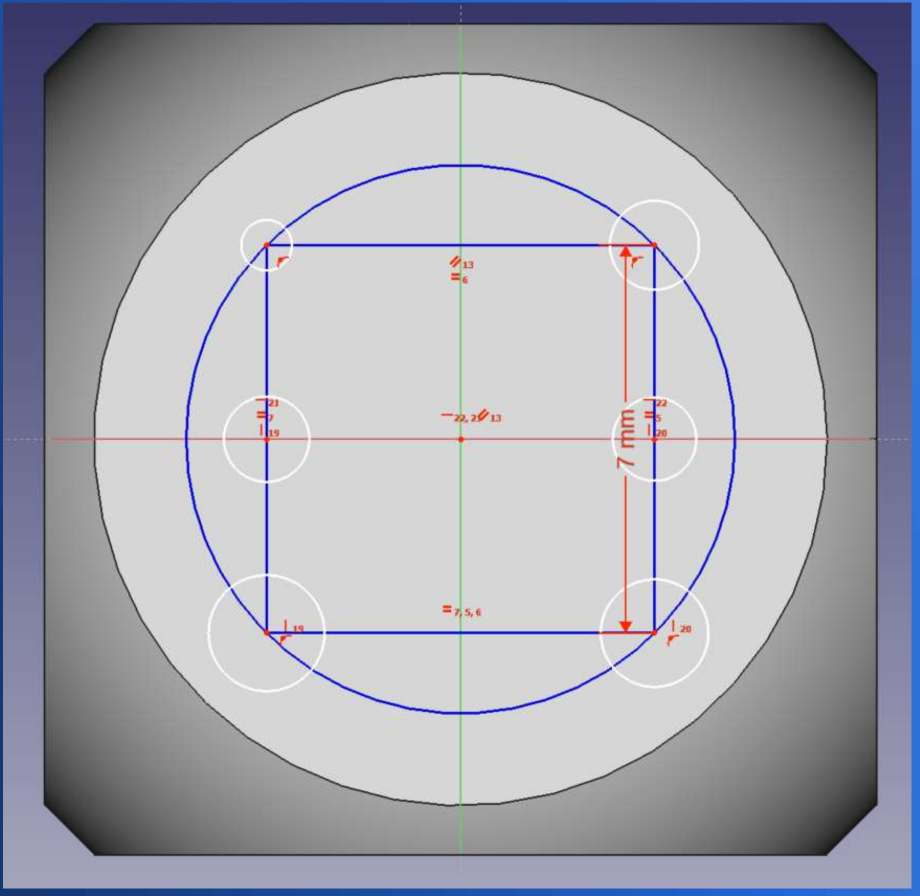
Project – Make a Dice

Now click on the Fix the radius of a circle

or an arc icon. Constraint_Radius

Set the radius of one of the circles to

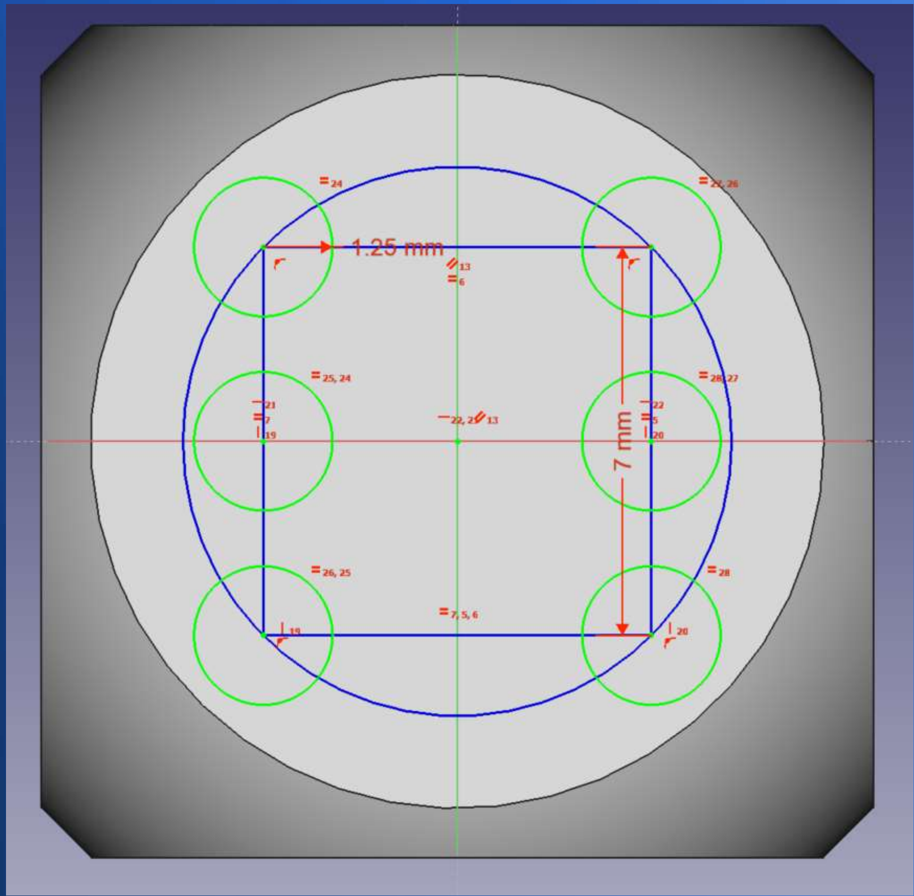
1.25mm.



Project – Make a Dice

All the circles should turn green, and a message in the Tasks tab should tell you that the sketch is fully constrained. Close the Tasks tab to return to the 3D view of your dice.

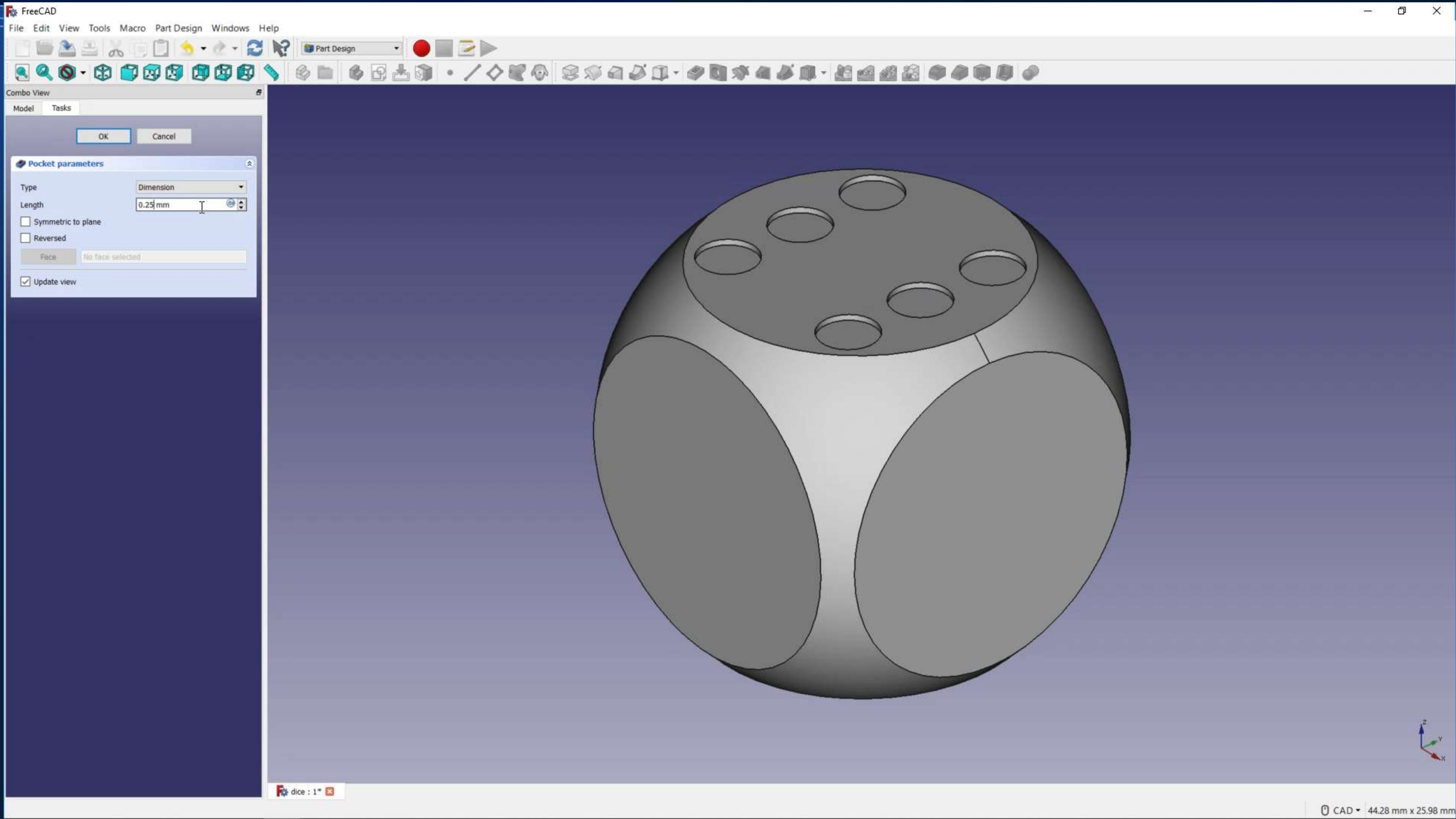
You should see the six circles on one side of the dice.



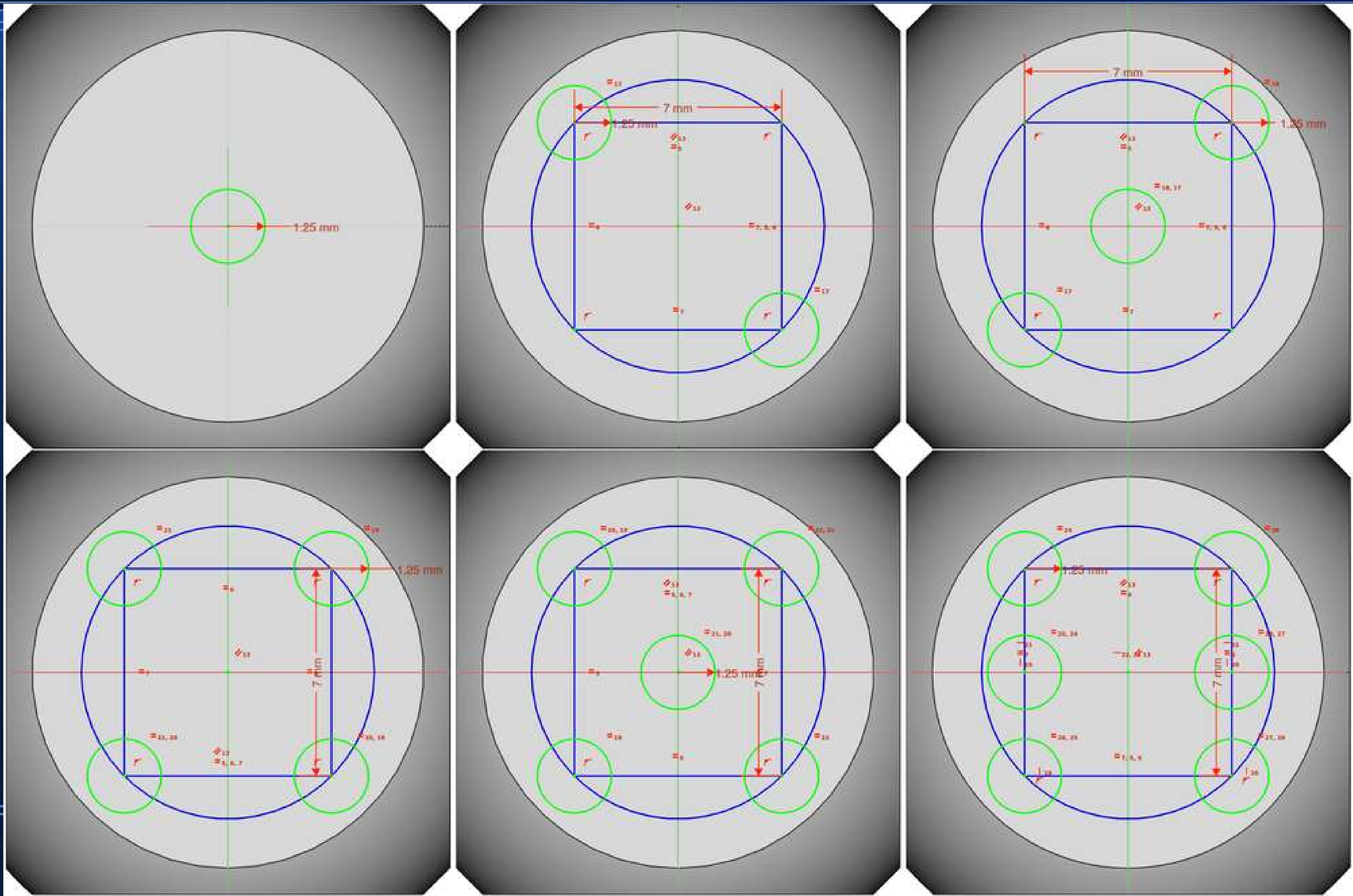
Project – Make a Dice

- Create pockets
- Now that you have drawn the dots, you need to set them into the surface the dice.
- Hold down the Ctrl key and select all six circles on the surface of the dice.
- Click on the Create a pocket with the selected sketch icon to create six pockets in the surface of the dice.
- Set the pocket depth to 0.25mm.
- Now you need to add dots to the other faces. Use exactly the same steps:
- Add construction geometry
- Draw circles and align them to the geometry
- Create pockets based on the circles as the dots

Project – Make a Dice



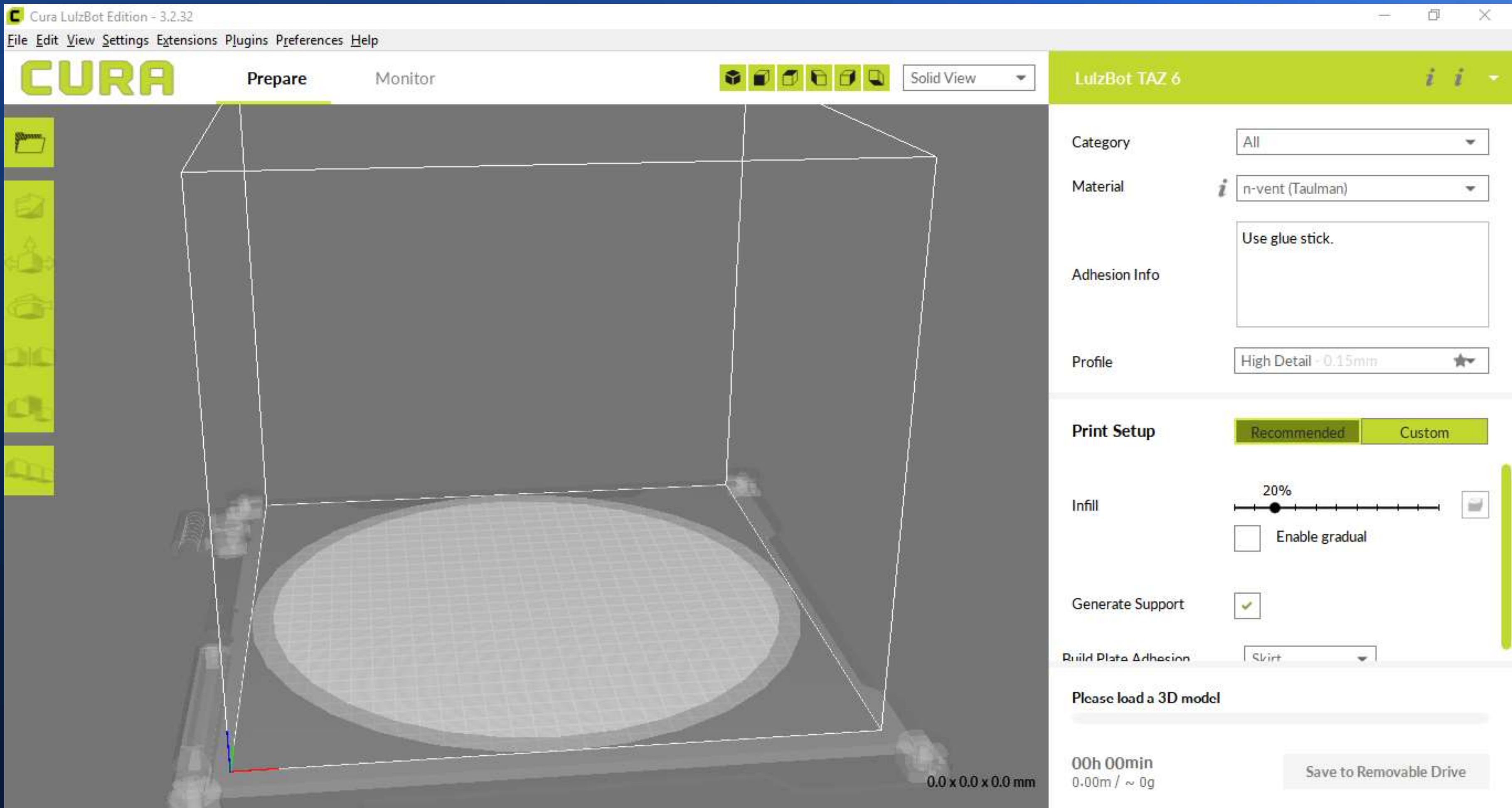
Project – Make a Dice



Project – Make a Dice

- Export your dice
- Now it’s time to export your dice model for printing. There are a lot of different types of 3D printers, but most of them accept .stl files or .gcode files. It is easiest to export your model as an .stl file and then use this file in another software program to create a .gcode file.
- In the Model tab, select the body.
- Click on File and then on Export.
- In the Files of type drop-down menu, click on the .stl option.
- Give your file a name, and then click on Save.
- If your 3D printer requires .gcode files, use another software program to convert the .stl file into a .gcode file. The instructions are for the software Cura.
- Open Cura from you application menu.

Project – Make a Dice



Project – Make a Dice

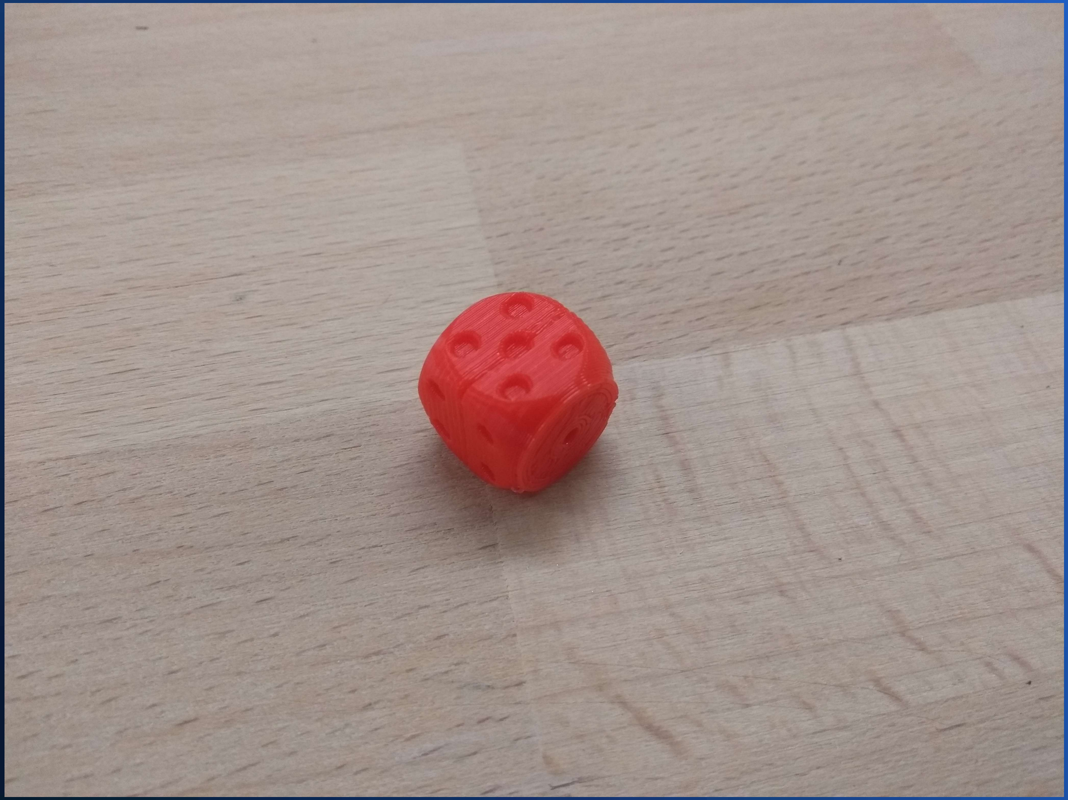
- If your 3D printer requires .gcode files, use another software program to convert the .stl file into a .gcode file. The instructions below are for the software Cura.
- Open Cura from you application menu.
- o to the File menu and open your dice.stl file.
- Use the settings menu on the right-hand side to choose your material, profile, infill, and any support you might need.
- In the example:
- The material is PLA
- The profile is set to High Detail
- The infill is set to 60% to give the dice a bit of weight.
- Once the settings are correct for your 3D printer, use the menu in the bottom left-hand corner to save your .gcode file.

Project – Make a Dice

- Print the dice!
- Instructions can’t be provided for all 3D printer models. The example here uses the Octoprint software to print the dice on a Lulzbot TAZ 6.
- Turn on the 3D printer and load the Octoprint software.
- Octoprint1 → Load your .gcode file so it appears it the file list.
- Octoprint2 →
- Then load and print the file.
- Octoprint3 →
- As soon as the printer is warmed up, it should start printing the dice.
- Printing:
- The dots on the dice may turn out a bit faint. If this is the case, you could:

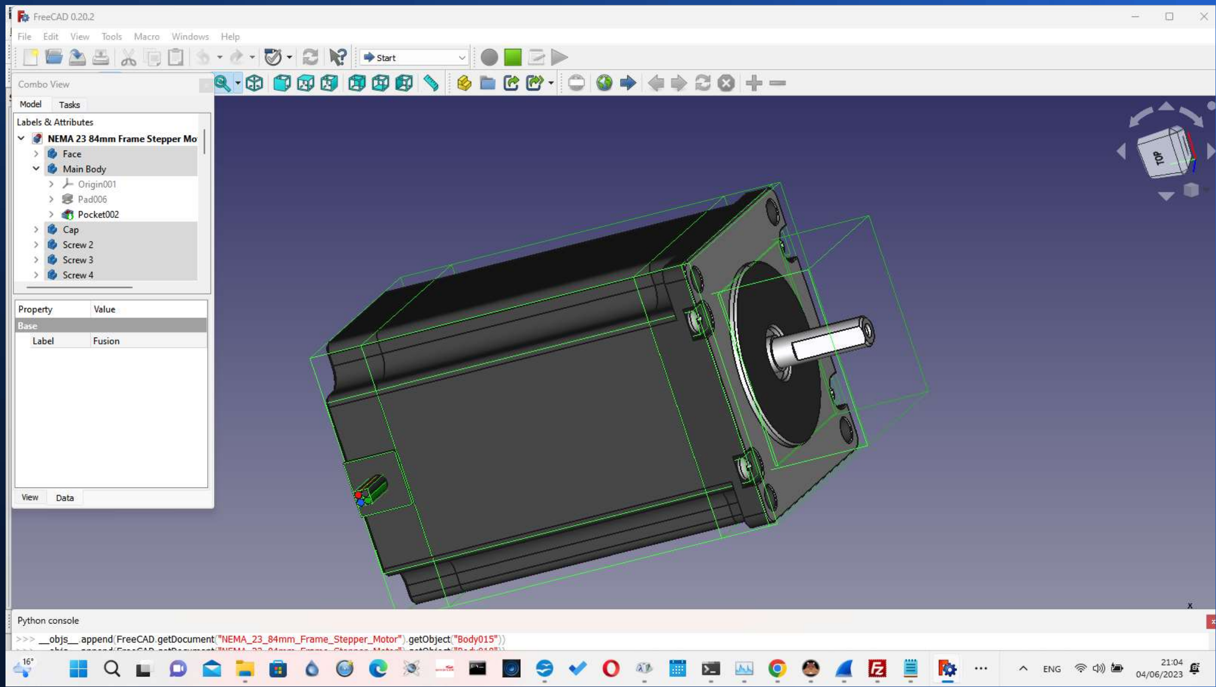
Project – Make a Dice

- Set the pocket size on your FreeCAD file to be a little deeper
- Simply use a permanent marker to colour in the dots!



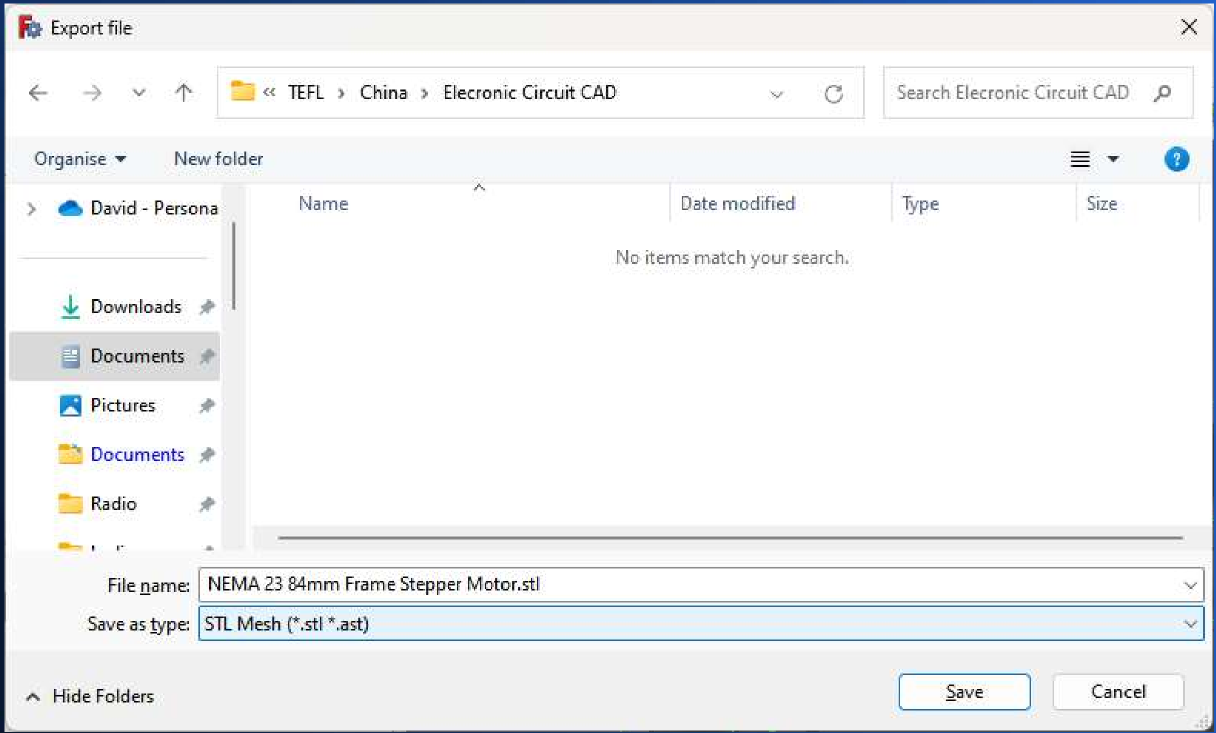
Freecad 3D printing Guide

- So you have a project ready to 3D print. But FreeCAD cannot produce .gcode files. So there ia a workaround. FreeCAD can export as a .stl file which Ultmaker Cura can import. It can make .gcode files.



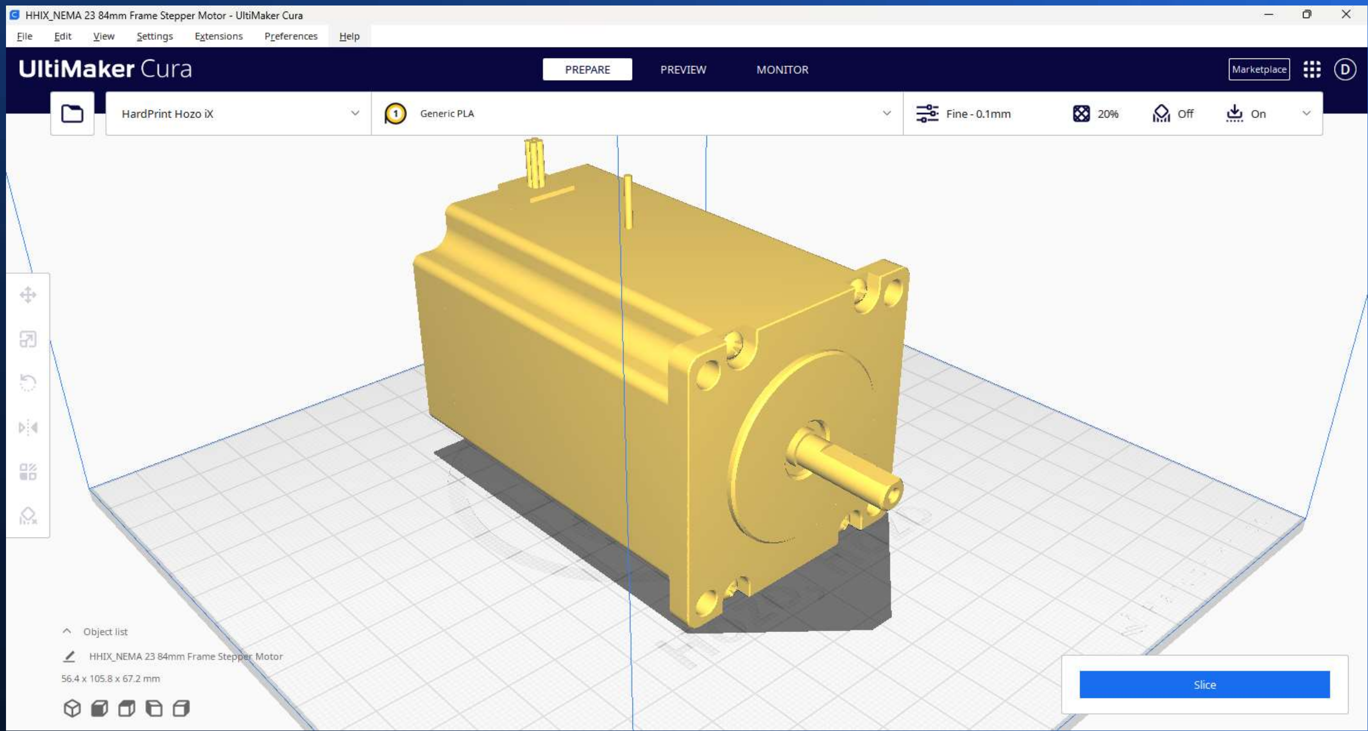
Freecad 3D printing Guide

- Export as .stl (STL mesh file). You can find both Untimaker Cura and FreeCAD for download here: http://dfdn.info/downloads
- This is my own site running on my own server in London.



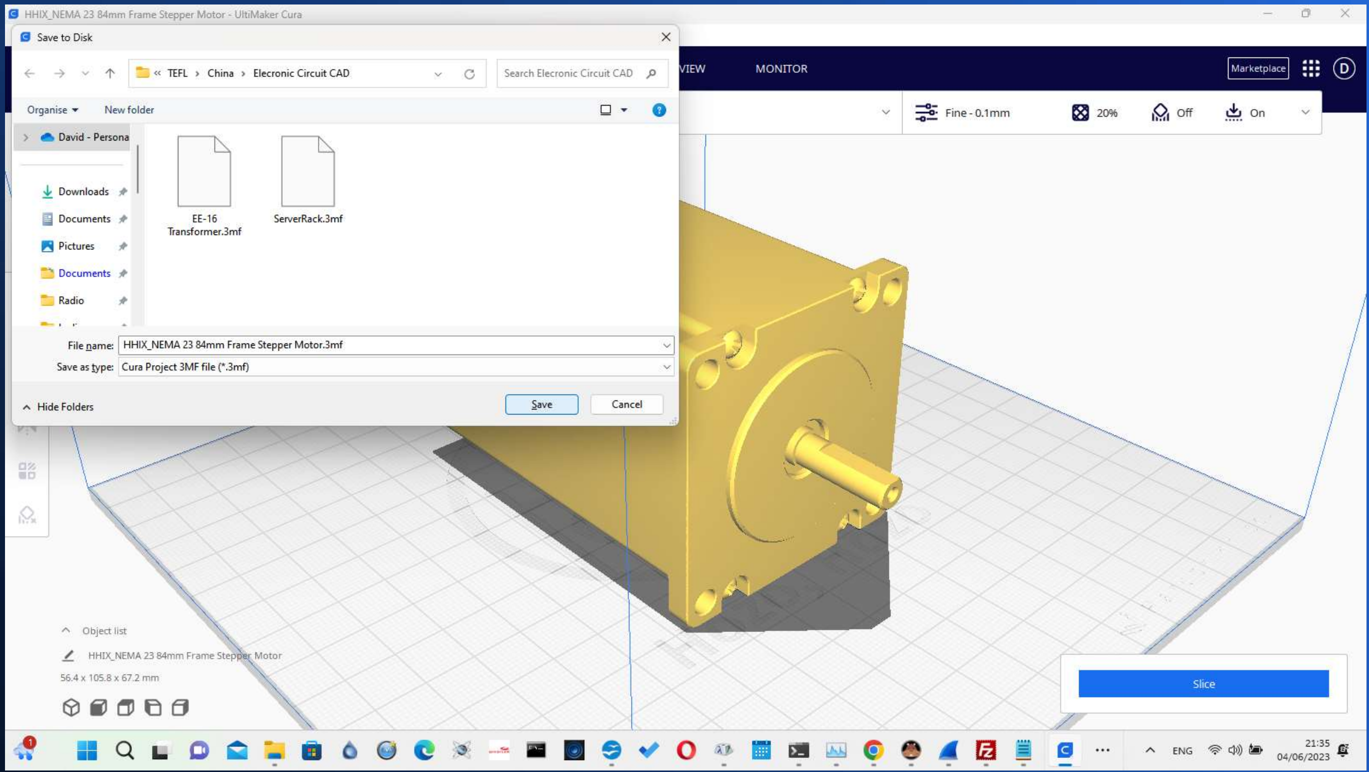
Freecad 3D printing Guide

- Once your .stl file is opened you can view from any angle. Now we are ready to export a .gcode file.



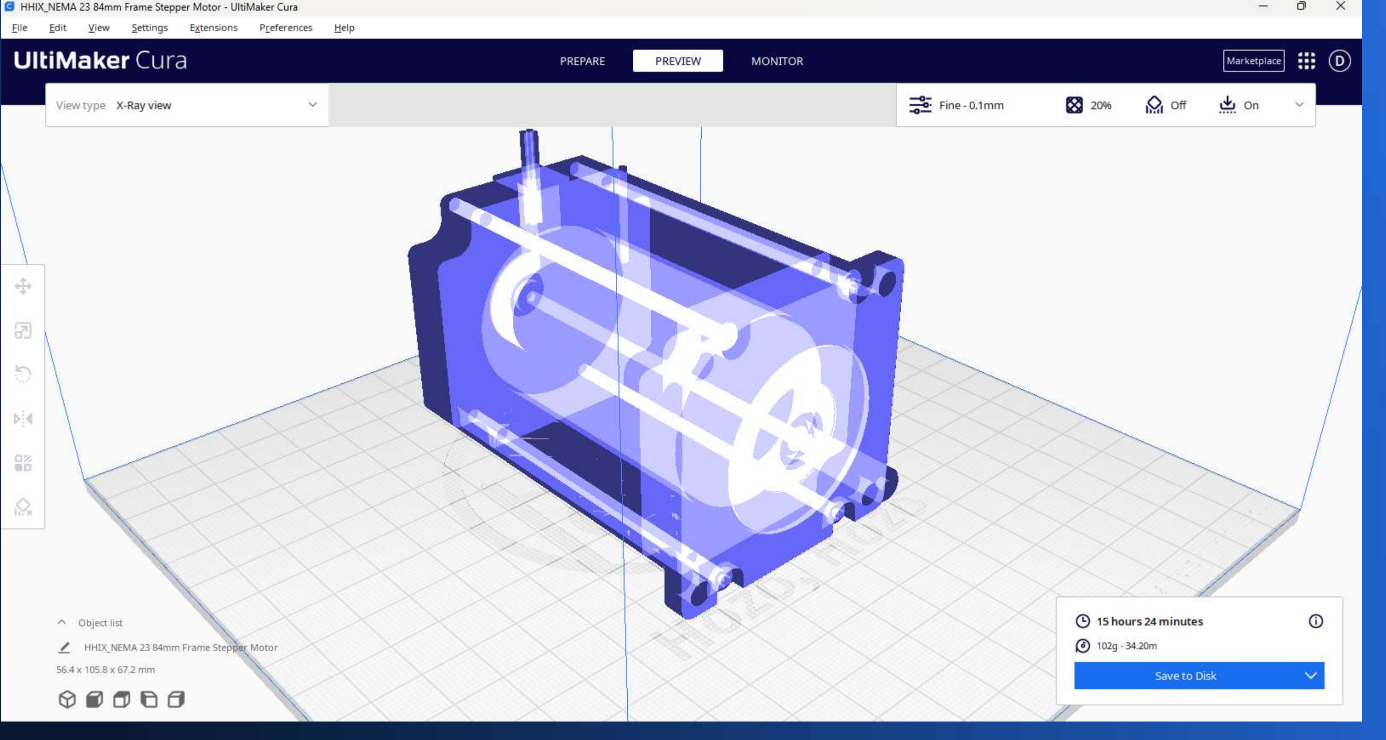
Freecad 3D printing Guide

- First save as a Cura Project 3MF file...



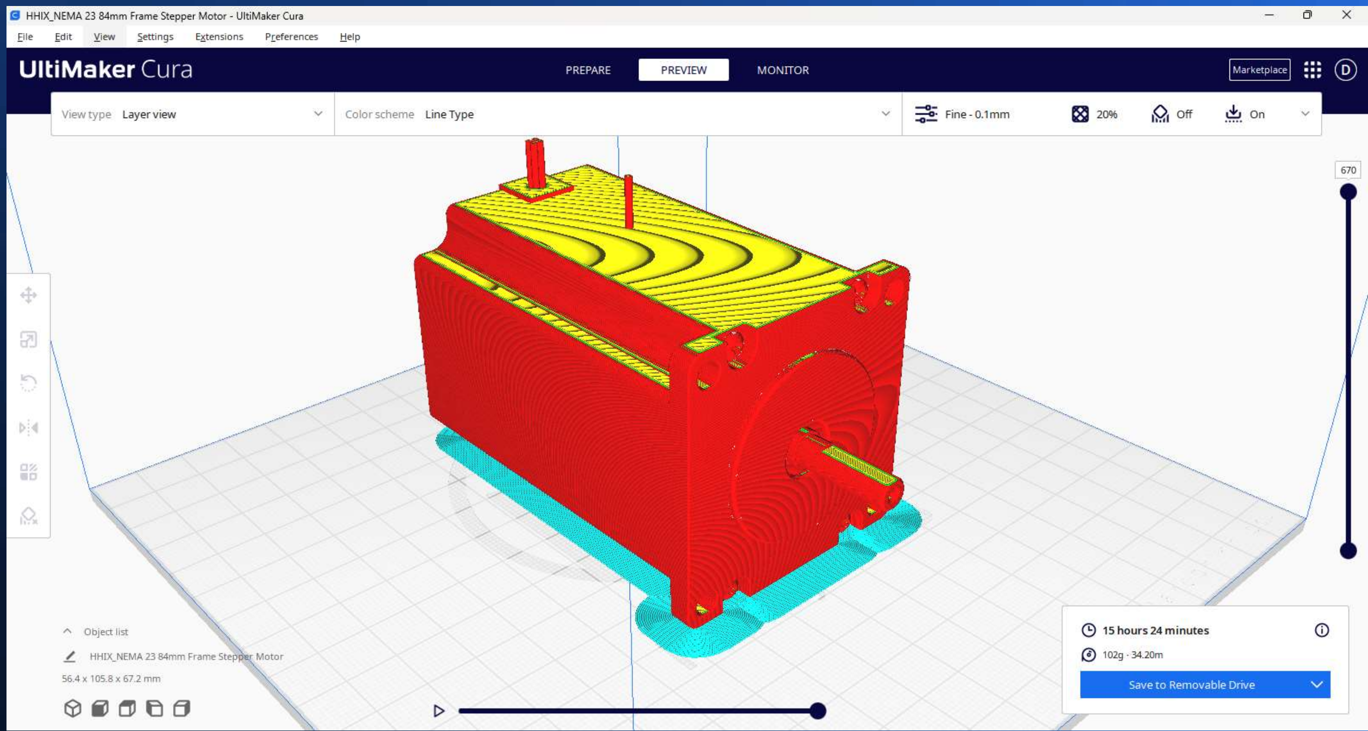
Freecad 3D printing Guide

- X-Ray View: - looks neat! Unmistakably 3D not 2D!



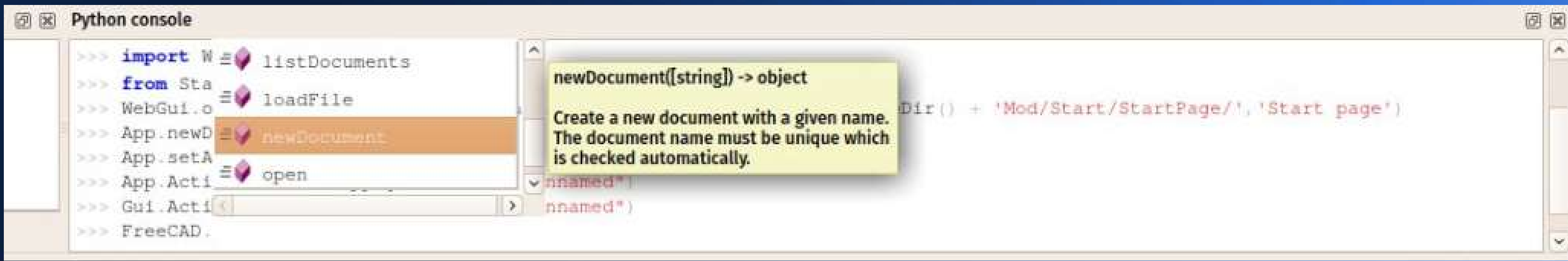
Freecad 3D printing Guide

- Now we slice the model before we can finally print it. Extimated time: 15 hours 24 minutes. Come back tomorrow... Job done.



Customising of CAD systems using macros and Application Programming Interface (API)

- Manipulating FreeCAD objects
- Let's start by creating a new empty document:
- doc = FreeCAD.newDocument()
- If you type this in the FreeCAD Python console, you will notice that as soon as you type "FreeCAD." (the word FreeCAD followed by a dot), a window pops up, allowing you to quickly autocomplete the rest of the line. Even better, each entry in the autocomplete list has a tooltip explaining what it does. This makes it very easy to explore the functionality available. Before choosing "newDocument", have a look at the other options available.



Customising of CAD systems using macros and Application Programming Interface (API)

- As soon as you press Enter our new document will be created. This is similar to pressing the "new document" button on the toolbar. In Python, the dot is used to indicate something that is contained inside something else (newDocument is a function that is inside the FreeCAD module). The window that pops up therefore shows you everything that is contained inside "FreeCAD". If you would add a dot after newDocument, instead of the parentheses, it would show you everything that is contained inside the newDocument function. The parentheses are mandatory when you are calling a Python function, such as this one. We will illustrate that better below.
- Now let's get back to our document. Let's see what we can do with it. Type the following and explore the available options:doc.
- Usually names that begin with an upper-case letter are attributes: they contain a value. Names that begin with a lower-case letter are functions (also called methods): they "do something". Names that begin with an underscore are usually there for the internal use of the module, and you should ignore them. Let's use one of the methods to add a new object to our document:
- box = doc.addObject("Part::Box","myBox")

Customising of CAD systems using macros and Application Programming Interface (API)

- Our box is added in the tree view, but nothing happens in the 3D view yet, because when working from Python, the document is never recomputed automatically. We must do that manually, whenever required:
- doc.recompute()
- Now our box has appeared in the 3D view. Many of the toolbar buttons that add objects in FreeCAD actually do two things: add the object, and recompute. Try now adding a sphere with the appropriate button in the Part Workbench, and you will see the two lines of Python code being executed one after the other.
- You can get a list of all possible object types like Part::Box:

Customising of CAD systems using macros and Application Programming Interface (API)

- doc.supportedTypes()
- Now let's explore the contents of our box:
- box.
- You'll immediately see a couple of very interesting things such as:
- box.Height
- This will print the current height of our box. Now let's try to change that:
- box.Height = 5
- If you select your box with the mouse, you will see that in the properties panel, under the Data tab, our Height property appears with the new value. All properties of a FreeCAD object that appear in the Data and View tabs are directly accessible by Python too, by their names, like we did with the Height property. Data properties are accessed directly from the object itself, for example:

Customising of CAD systems using macros and Application Programming Interface (API)

- box.Length
- View properties are stored inside a ViewObject. Each FreeCAD object possesses a ViewObject, which stores the visual properties of the object. When running FreeCAD without its Graphical Interface (for example when launching it from a terminal with the -c command line option, or using it from another Python script), the ViewObject is not available, since there is no visual at all.
- Try the following example to access the line color of our box:
- box.ViewObject.LineColor

Customising of CAD systems using macros and Application Programming Interface (API)

- Vectors and Placements
- Vectors are a fundamental concept in any 3D application. It is a list of 3 numbers (x, y and z), describing a point or position in the 3D space. A lot of things can be done with vectors, such as additions, subtractions, projections and much more. In FreeCAD vectors work like this:
- myvec = FreeCAD.Vector(2,0,0)
- print(myvec)
- print(myvec.x)
- print(myvec.y)
- othervec = FreeCAD.Vector(0,3,0)
- sumvec = myvec.add(othervec)

Customising of CAD systems using macros and Application Programming Interface (API)

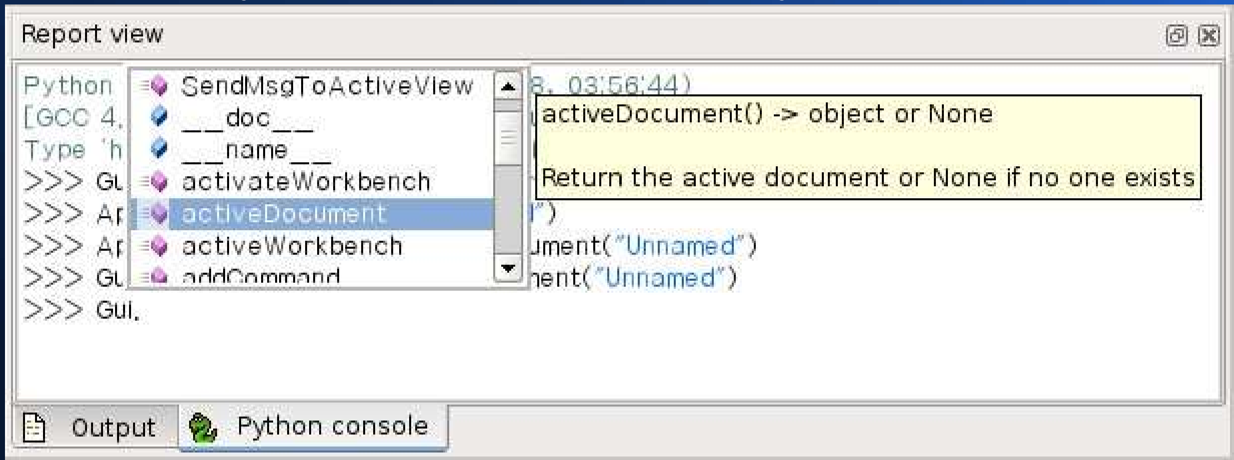
- Another common feature of FreeCAD objects is their Placement. As we saw in earlier sections, each object has a Placement property, which contains the position (Base) and orientation (Rotation) of the object. These properties are easy to manipulate from Python, for example to move our object
- print(box.Placement)
- print(box.Placement.Base)
- box.Placement.Base = sumvec
- otherpla = FreeCAD.Placement()
- otherpla.Base = FreeCAD.Vector(5,5,0)
- box.Placement = otherpla

Customising of CAD systems using macros and Application Programming Interface (API)

- Let's start by creating a new empty document:
- doc = FreeCAD.newDocument()
- If you type this in the FreeCAD Python console, you will notice that as soon as you type FreeCAD. a window pops up, allowing to quickly autocomplete the rest of your line. Even better, each entry in the autocomplete list has a tooltip explaining what it does. This makes it easier to explore the available functionality. Before choosing newDocument, have a look at the other options.
- Now our new document will be created. This is similar to pressing the Std New.svg Std New button on the toolbar. In fact most buttons in FreeCAD do nothing more than execute one or more lines of Python code. Even better, you can set an option in Edit → Preferences → General → Macro to Show script commands in python console. This will print in the console all Python code executed when you press buttons. Very useful for learning how to reproduce actions in Python.
- Now let's get back to our document and see what we can do with it:

Customising of CAD systems using macros and Application Programming Interface (API)

- The autocomplete mechanism of the FreeCAD Python console:
- box = doc.addObject("Part::Box", "myBox")
- Nothing happens. Why? Because FreeCAD is made for the big picture. One day, it will work with hundreds of complex objects, all depending each other. Making a small change somewhere could have a big impact; you may need to recalculate the whole document which could take a long time. For that reason almost no command updates the scene automatically. You must do it manually:



Customising of CAD systems using macros and Application Programming Interface (API)

- doc.recompute()
- Now our box appeared. Many of the buttons that add objects in FreeCAD actually do two things: add the object, and recompute. If you turned on the Show script commands in python console option above, try adding a sphere with the GUI button; you'll see the two lines of Python code being executed one after the other.
- Now let's explore the contents of our box:
- box.
- You'll immediately see a couple of very interesting things such as:
- box.Height
- This will print the current height of our box. Now let's try to change that:
- box.Height = 5

Customising of CAD systems using macros and Application Programming Interface (API)

- If you select your box with the mouse, you'll see that in the Property editor, on the Data tab, our DataHeight property appears. All properties of a FreeCAD object that appear there (and also on the View tab, more about that later), are directly accessible in Python too, by their names, like we did with the DataHeight property. Try changing the other dimensions of the box.
- Vectors and placements
- Vectors are a very fundamental concept in any 3D application. A vector is a list of 3 numbers (x, y and z), describing a point or position in 3D space. Many things can be done with vectors, such as additions, subtractions, projections and much more. In FreeCAD vectors work like this:
- myvec = FreeCAD.Vector(2, 0, 0)
- myvec.x
- myvec.y
- othervec = FreeCAD.Vector(0, 3, 0)

Customising of CAD systems using macros and Application Programming Interface (API)

- sumvec = myvec.add(othervec)
- Another common feature of FreeCAD objects is their placement. Each object has a DataPlacement property, which contains the DataBase (position) and DataRotation (orientation) of the object. It is easy to manipulate, for example to move our object:
- box.Placement
- box.Placement.Base
- box.Placement.Base = sumvec
- otherpla = FreeCAD.Placement()
- box.Placement = otherpla
- Now you must understand a couple of important concepts before we get further.

Customising of CAD systems using macros and Application Programming Interface (API)

- App and Gui
- FreeCAD has been designed so that it can also be used without its user interface, as a command-line application. Almost every object in FreeCAD therefore consists of two parts: an Object, its "geometry" component, and a ViewObject, its "visual" component. When you work in command-line mode, the geometry part is present, but the visual part is disabled.
- To illustrate the concept let's look at our cube object. The geometric properties of the cube, such as its dimensions, position, etc. are stored in the Object. While its visual properties, such as its color, line thickness, etc. are stored in the ViewObject. This corresponds to the Data and View tabs in the Property editor. The view object of an object is accessed like this:
- vo = box.ViewObject
- Now you can also change the properties on the View tab:
- vo.Transparency = 80
- vo.hide()
- vo.show()

Customising of CAD systems using macros and Application Programming Interface (API)

- When you start FreeCAD, the Python console already loads two base modules: FreeCAD and FreeCADGui (which can also be accessed by their shortcuts App and Gui). They contain all kinds of generic functionality to work with documents and their objects. To illustrate our concept, see that both FreeCAD and FreeCADGui contain an ActiveDocument attribute, which is the currently opened document. FreeCAD.ActiveDocument and FreeCADGui.ActiveDocument are not the same object however. They are the two components of a FreeCAD document, and they contain different attributes and methods. For example, FreeCADGui.ActiveDocument contains ActiveView, which is the currently opened 3D view.

Customising of CAD systems using macros and Application Programming Interface (API)

- Modules
- The true power of FreeCAD lies in its faithful modules, with their respective workbenches. The FreeCAD base application is more or less an empty container. Without its modules it can do little more than create new, empty documents. Each module not only adds new workbenches to the interface, but also new Python commands and new object types. As a result several different, and even totally incompatible, object types can coexist in the same document. The most important modules in FreeCAD that we'll look at in this tutorial are: Part, Mesh, Sketcher and Draft.
- Sketcher and Draft both use the Part module to create and handle their geometry. While Mesh is totally independent, and handles its own objects. More about that below.
- You can check all the available base object types for the current document like this:
- doc.supportedTypes()

Customising of CAD systems using macros and Application Programming Interface (API)

- The different FreeCAD modules are not automatically loaded in the Python console. This is to avoid having a very slow startup. Modules are loaded only when you need them. So, for example, to explore what's inside the Part module:
- import Part
- Part.
- But we'll talk more about the Part module next.

Customising of CAD systems using macros and Application Programming Interface (API)

- Mesh module
- Meshes are a very simple kind of 3D object, used for example by Sketchup, Blender and 3D Studio Max. They are composed of 3 elements: points (also called vertices), lines (also called edges) and faces. In many applications, FreeCAD included, faces can have only 3 vertices. Of course, nothing prevents you from having a bigger face made up of several coplanar triangles.
- Meshes are simple, but because they are simple you can easily have millions of them in a single document. However, in FreeCAD they have less use and are mostly there so you can import objects in mesh formats (.stl, .obj) from other applications. The Mesh module was also used extensively as the main test module in the first month of FreeCAD's life.
- Mesh objects and FreeCAD objects are different things. You can see the FreeCAD object as a container for a Mesh object (and as we'll see below, for Part objects also). So in order to add a mesh object to FreeCAD, we must first create a FreeCAD object and a Mesh object, then add the Mesh object to the FreeCAD object:

Customising of CAD systems using macros and Application Programming Interface (API)

- import Mesh
- mymesh = Mesh.createSphere()
- mymesh.Facets
- mymesh.Points
- meshobj = doc.addObject("Mesh::Feature", "MyMesh")
- meshobj.Mesh = mymesh
- doc.recompute()
- This is a standard example that uses the createSphere() method to create a sphere, but you can also create custom meshes from scratch by defining their vertices and faces.
- For more about mesh scripting: https://wiki.freecad.org/Mesh_Scripting

Customising of CAD systems using macros and Application Programming Interface (API)

- Part module
- The Part module is the most powerful module in the whole of FreeCAD. It allows you to create and manipulate BRep objects. BREP stands for "Boundary Representation". A BREP object is defined by surfaces that enclose and define an inner volume. Unlike meshes, BREP objects can have a wide variety of components from planar faces to very complex NURBS surfaces.
- The Part module is based on the powerful OpenCasCade library, which allows a wide range of complex operations to be performed on those objects, such as boolean operations, filleting, lofts, etc.
- The Part module works the same way as the Mesh module: You create a FreeCAD object, a Part object, then add the Part object to the FreeCAD object:
- import Part
- myshape = Part.makeSphere(10)
- myshape.Volume

Customising of CAD systems using macros and Application Programming Interface (API)

- myshape.Area
- shapeobj = doc.addObject("Part::Feature", "MyShape")
- shapeobj.Shape = myshape
- doc.recompute()
- The Part module (like the Mesh module) also has a shortcut that automatically creates a FreeCAD object and adds a shape to it, so you can shorten the last three lines to:
- Part.show(myshape)
- By exploring the contents of myshape, you will notice many interesting subcomponents such as Faces, Edges, Vertexes, Solids and Shells, and a wide range of geometry operations such as cut (subtraction), common (intersection) or fuse (union). The Topological data scripting page explains all that in detail.
- Read more about part scripting... https://wiki.freecad.org/Topological_data_scripting

Customising of CAD systems using macros and Application Programming Interface (API)

- Draft module
- FreeCAD features many more modules, such as Sketcher and Draft, that also create Part objects. These modules add additional parameters to the objects created, or even implement a whole new way to handle the Part geometry in them. Our box example above is a perfect example of a parametric object. All you need to define the box is to specify the parameters height, width and length. Based on those, the object will automatically calculate its Part shape. FreeCAD allows you to create such objects in Python.
- The Draft module adds 2D parametric object types (which are all Part objects) such as lines and circles, and also provides some generic functions that not only work on Draft objects, but on any Part object. To explore what is available, simply do:
- import Draft
- rec = Draft.makeRectangle(5, 2)
- mvec = FreeCAD.Vector(4, 4, 0)

Customising of CAD systems using macros and Application Programming Interface (API)

- Draft.move(rec, mvec)
- Draft.move(box, mvec)
- **Interface:**
- The FreeCAD user interface is made with Qt, a powerful graphical interface system, responsible for drawing and handling all the controls, menus, toolbars and buttons around the 3D view. Qt provides a module, PySide, which allows Python to access and modify Qt interfaces such as FreeCAD's. Let's try to fiddle with the Qt interface and produce a simple dialog:
- from PySide import QtGui
- QtGui.QMessageBox.information(None, "Apollo program", "Houston, we have a problem")
- Notice that the dialog that appears has the FreeCAD icon in its toolbar, meaning that Qt knows that the order has been issued from inside the FreeCAD application. It is possible to manipulate any part of the FreeCAD interface.

Customising of CAD systems using macros and Application Programming Interface (API)

- Qt is a very powerful interface system that allows you to do very complex things. It also has some easy-to-use tools such as the Qt Designer with which you can design dialogs graphically and then add them to the FreeCAD interface with a few lines of Python code.
- Read more about PySide here... https://wiki.freecad.org/PySide
- Macros
- Now that you have a good understanding of the basics, where are we going to keep our Python scripts, and how are we going to launch them inside FreeCAD? There is an easy mechanism for that, called Macros. A macro is a Python script that can be added to a toolbar and launched via a mouse click. FreeCAD provides you with a simple text editor (Macro → Macros... → Create) where you can write or paste scripts. Once the script is done, use Tools → Customise... → Macros to define a button for it that can be added to toolbars.
- Now you are ready for more in-depth FreeCAD scripting. So head on to the Power users hub! https://wiki.freecad.org/Power_users_hub

Customising of CAD systems using macros and Application Programming Interface (API)

- Qt is a very powerful interface system that allows you to do very complex things. It also has some easy-to-use tools such as the Qt Designer with which you can design dialogs graphically and then add them to the FreeCAD interface with a few lines of Python code.
- Read more about PySide here... https://wiki.freecad.org/PySide
- Macros
- Now that you have a good understanding of the basics, where are we going to keep our Python scripts, and how are we going to launch them inside FreeCAD? There is an easy mechanism for that, called Macros. A macro is a Python script that can be added to a toolbar and launched via a mouse click. FreeCAD provides you with a simple text editor (Macro → Macros... → Create) where you can write or paste scripts. Once the script is done, use Tools → Customise... → Macros to define a button for it that can be added to toolbars.
- Now you are ready for more in-depth FreeCAD scripting. So head on to the Power users hub! https://wiki.freecad.org/Power_users_hub

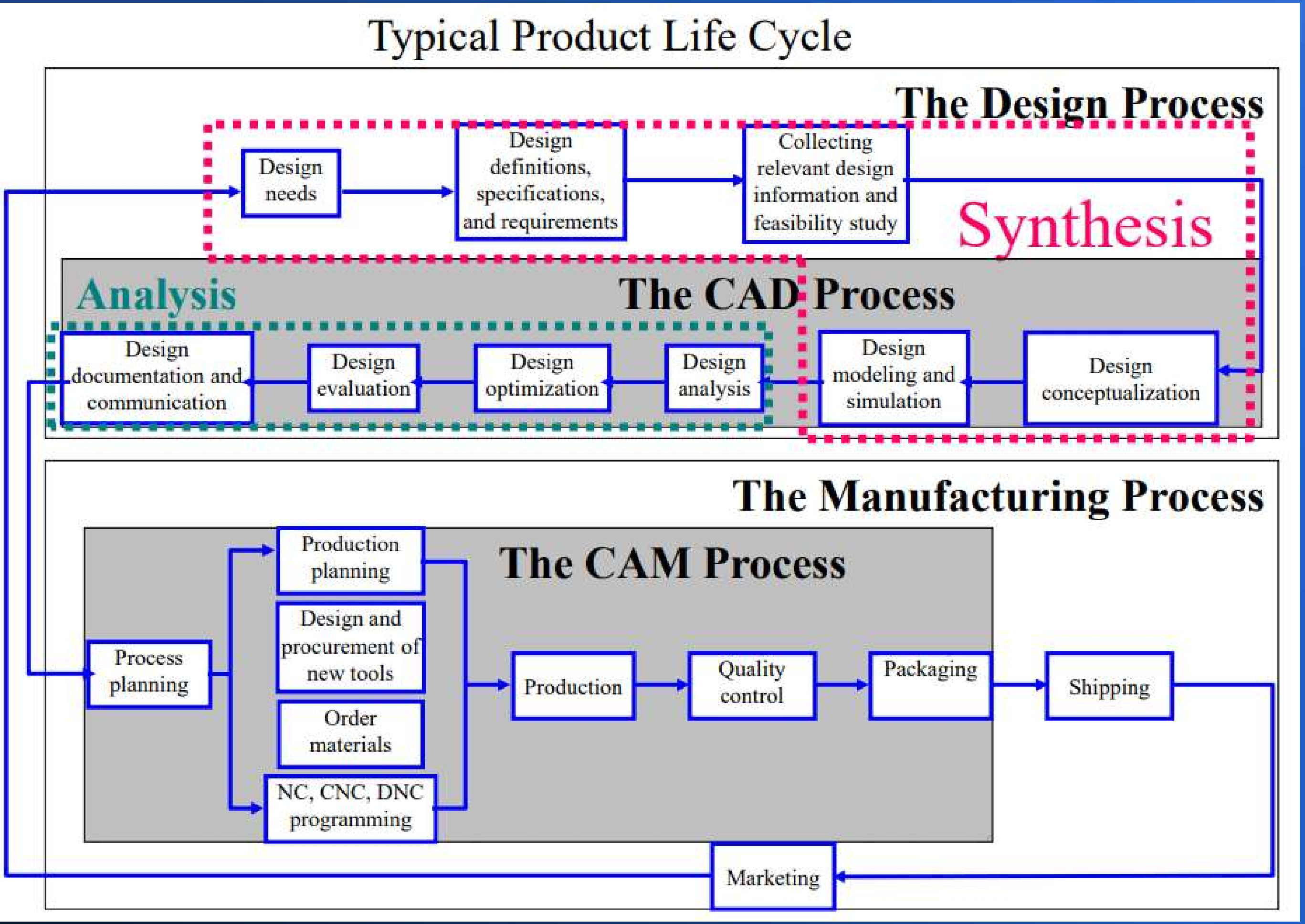
Customising of CAD systems using macros and Application Programming Interface (API)

- For a demonstration of the application program interface {API} and macro programming:
- http://dfdn.info/video/FreeCAD-Sketching-Demonstration/FreeCAD-Sketching-Demonstration.html
- Made with Camtasia Studio, hosted by Apache webserver on a Raspberry PI server.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- CAD stands for computer-aided design
- CAM stands for computer-aided manufacturing;
- CNC stands for computer numerical control;
- CAE is the use of computer simulation to analyse physical problems;
- Mastercam, for example, is a software that provides both CAD and CAM functionality to drive CNC machines efficiently for optimised productivity.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer



Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- What are the Functions of the CAD System?
- Reuse of design components.
- Ease of design modification and versioning.
- Automatic generation of standard components of the design.
- Validation/verification of designs against specifications and design rules.
- Simulation of designs without building a physical prototype.
- Automatic design of assemblies.
- The output of engineering documents, such as manufacturing drawings, and bills of materials.
- Direct output of designs to production units.
- Direct output to rapid prototyping or rapid industrial prototyping machines.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- What is Computer-Aided Engineering (CAE)?
- Computer-Aided Engineering (CAE) is mainly used for simulation analysis, verification, and improvement of designs. With the rapid development of computers and 3D CAD in recent years, the proportion of CAE applications has become higher and higher, the difficulty of use has become lower and lower, and the number of users has grown significantly. Compared to CAD, CAE users require more physics-related knowledge to set conditions and interpret results. As for the mathematical methods used in the calculation process, such as differential equations, finite element method, finite volume method, etc., all can be handed over to the computer for processing.
- Advantages of CAE:
- Save experiment cost and time, and speed up the development process.
- Physical quantities that are difficult to obtain experimentally can be obtained through simulation.
- It is easier to observe physical phenomena and changes in physical quantities.
- Product performance can be further optimised through numerical methods.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Disadvantages of CAE:
- Large-scale computing often requires expensive high-performance equipment.
- The interpretation of simulation results still requires the experience and knowledge of engineers to make correct judgments.
- What are the Basic Processes of CAE?
- Step 0: Geometric model processing
- The 0th step is the processing of the geometric model. Why is it the 0th step? One is that this step will not use the simulation software. Mainly because the processing quality will greatly affect the solution, and sometimes it is quite cumbersome to handle. For example, in the fluid simulation, the inner space of the fluid flow is required, and the inner space is not especially drawn in general, so the inner space must be drawn according to the shell geometry. In addition, the complex features inside the model should also be simplified as much as possible. Although the pipeline has no complex features, if the model is not properly simplified, it will consume a lot of resources to calculate, and even the solution will fail.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Step 1: Preprocessing
- The pre-processing is mainly to divide the mesh and set the simulation conditions. Usually, the geometry of the simulated object is quite complex, and there is no way to directly calculate the Governing equation corresponding to this geometry. To solve this problem, researchers cut this complex shape into multiple simple elements, which may be cubes, polyhedrons, etc., and then solve the equation on each element, and then obtain the corresponding value of the entire complex shape through mathematical operations. Simulation results. This step of slicing complex shapes into multiple elements is meshing. Continue to set simulation conditions, generally referred to as boundary conditions. In addition, the condition set also includes the selection of different physical models and the selection of media.
- Step 2: Solve
- After completing the simulation settings, in the solution step, you need to tell the software what method to use to calculate the problem, including the grid discretisation format, the selection of the solution algorithm, and so on. After completing the solution set, the next step is to wait for the software or the program written by yourself to solve it.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Step 3: Post-processing
- Assuming everything went well and the software successfully calculated the answer, the next step is to look at the results. Use post-processing software to view information such as velocity fields, flow traces, etc. inside the pipeline. If it is necessary to improve and optimise the product, it can also be judged from the simulation results, and then the relevant configuration and simulation conditions can be improved.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- What is Computer-Aided Manufacturing (CAM)?
- Computer-Aided Manufacturing (CAM) is the process of manufacturing product components in which engineers make extensive use of product lifecycle management computer software. In other words, CAM is used to assist manufacturing, and the final output of CAM automatic programming is the CNC machining program. The 3D models of components generated in CAD are used to generate the CNC code that drives the numerically controlled machine tools. This includes the engineer selecting the type of tool, the machining process, and the machining path.
- The Processing Steps of Computer-Aided Manufacturing:
- Depending on the material and the software used, each step is a simple or complex strategy.
- Rough machining:
- The process begins with cubic stock or castings, which are roughly machined into the final model. Because it is processed in the horisontal direction, a stepped shape is usually obtained. Common strategies are zig-zag clearing, offset clearing, plunge roughing, and rest-roughing.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Semi-finishing:
- The process starts with a roughed, uneven part, and the part is machined at a fixed offset.
- Finishing:
- Like semi-finishing, but with different starting materials.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Changes brought by CAD/CAM technology to industrial manufacturing:
- CAD/CAM refers to the use of computers to analyze, simulate, design, draw and draw up production plans, and manufacturing procedures, and control the production process, that is, from design to processing, all relying on the help of computers, so CAD/CAM is an important part of automation, affecting industrial productivity and quality.
- The production of CNC machining:
- With the growth of the production of consumer goods in the aviation industry, the automobile industry, and the light industry, the requirements for precision of products have increased, the structure and shape have become more complex, and the development and manufacturing cycle has become shorter and shorter, and traditional machining cannot meet the demand, so CNC machining came into being.
- The generation of CAD/CAM software:
- The application of CNC machining improves the production accuracy and production efficiency. However, when the structure and shape of the product are complex, it is difficult to complete the preparation of the machining program by using the traditional numerical control manual programming technology, so CAD/CAM emerges as the times require.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Features of CAD/CAM software:
- Three-dimensional modeling function; as mentioned above, the geometric information of the machined surface is the basis for the calculation of the software tool path, so the software can provide basic surface modeling functions.
- Parameter management: parameters include processing objects, tool parameters, machine tool parameters, processing technology, etc.; parameter setting is the main operation content of interactive graphics programming. It includes input modification of parameters, management optimisation, etc.
- Policy:
- Intuitively and realistically graphically simulate the machining process to verify whether there is any problem with the programmed program.
- Editing and modification of tool path:
- Provide various methods to edit the NC tool path.

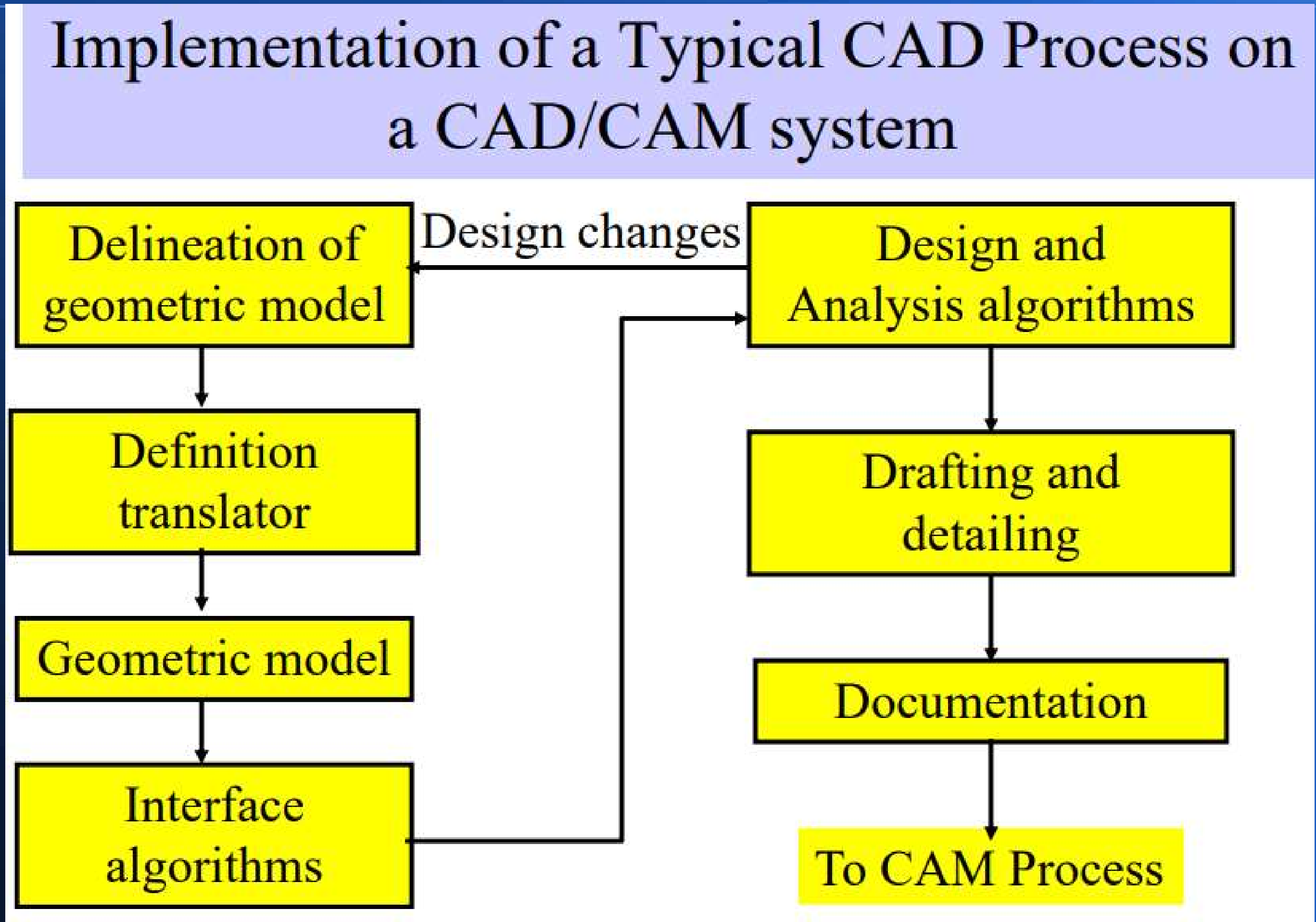
Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Post-processing:
- It is a word processing process, the program can meet the equipment requirements, and some auxiliary instructions are added at the beginning and end of the program.
- Process document generation:
- Write the information required by the machine tool operator (such as program name, machining sequence, and tool data) into standard and standardised documents.

Applications and Analysis of CAD in Product Life Cycle:
CAD/CAM/CNC/CAE, and data transfer

- The product begins with a need which is identified
- based on customers' and markets' demands.
- • The product goes through two main processes from
- the idea conceptualisation to the finished product:
- 1. The design process.
- 2. The manufacturing process.
- The main sub-processes that constitute the design process
- are:
- 1. Synthesis.
- 2. Analysis.

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

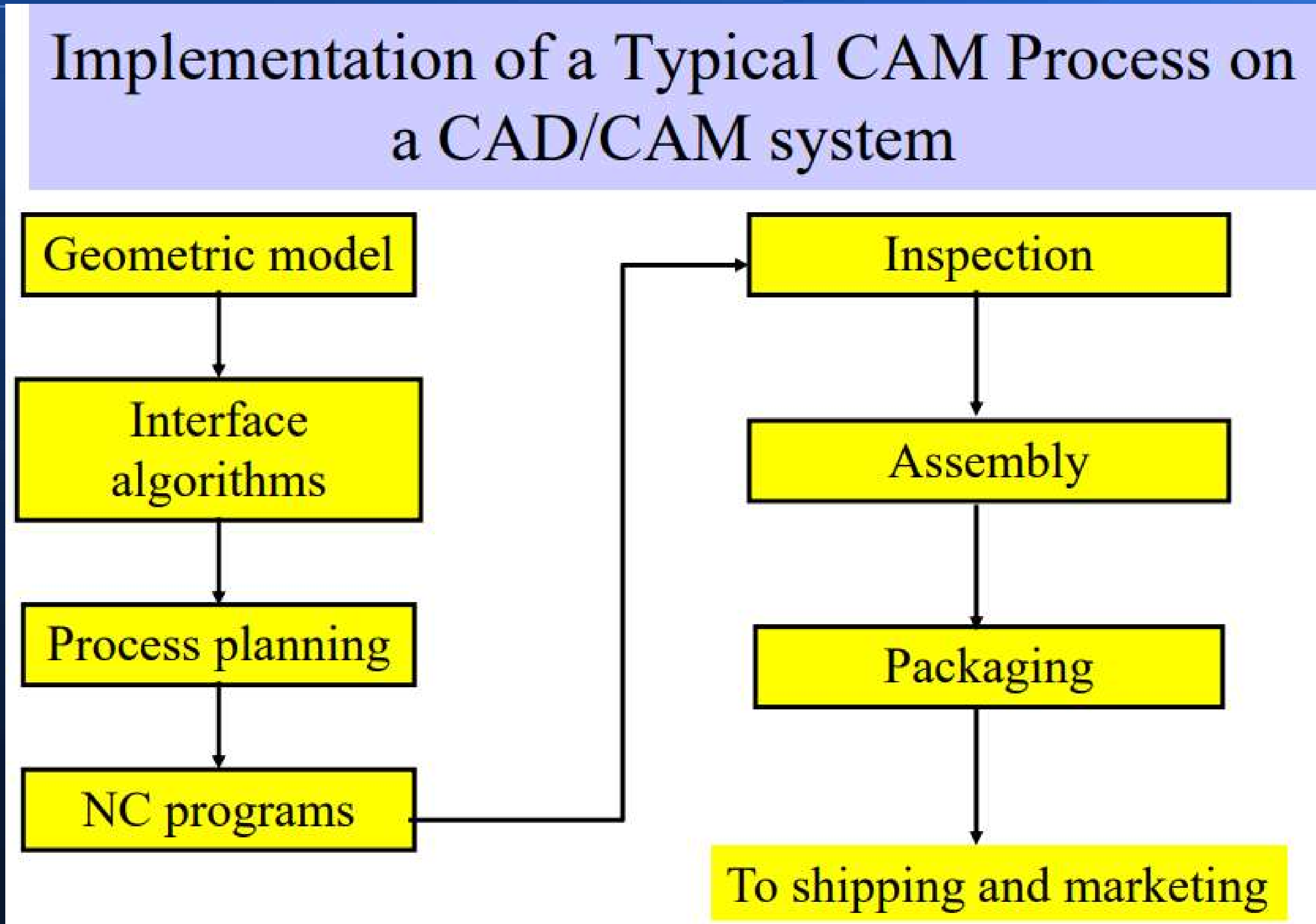


Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

CAD Tools Required to Support the Design Process

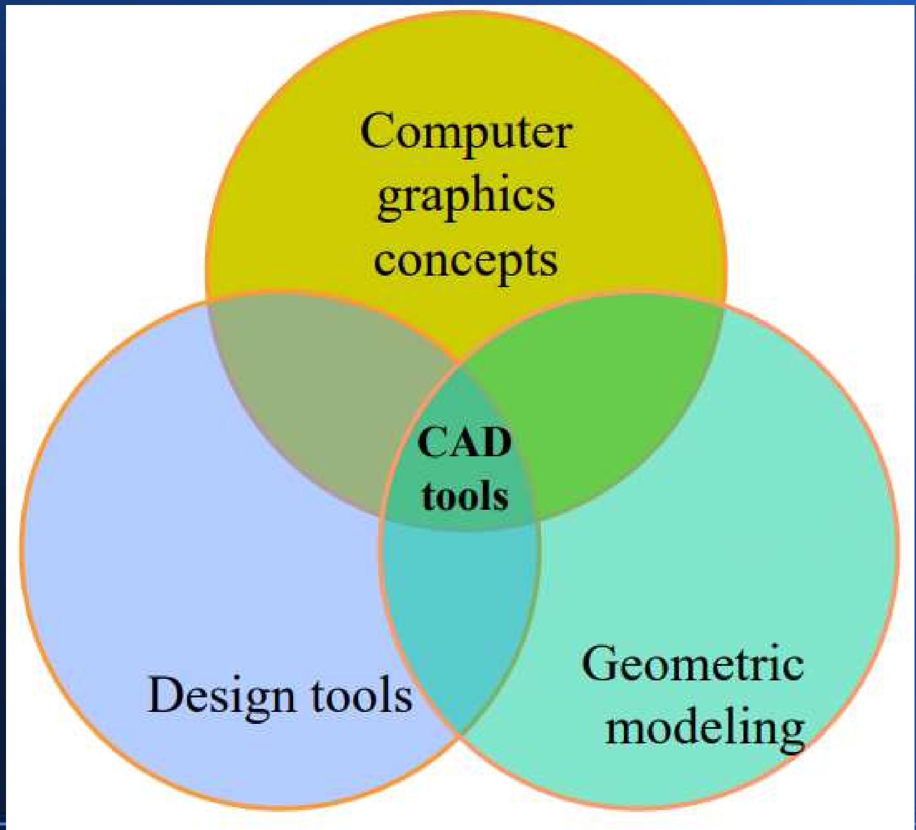
Design phase	Required CAD tools
Design conceptualization	Geometric modeling techniques; Graphics aids; manipulations; and visualization
Design modeling and simulation	Same as above; animation; assemblies; special modeling packages.
Design analysis	Analysis packages; customized programs and packages.
Design optimization	Customized applications; structural optimization.
Design evaluation	Dimensioning; tolerances; BOM; NC.
Design communication and documentation	Drafting and detailing…

Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

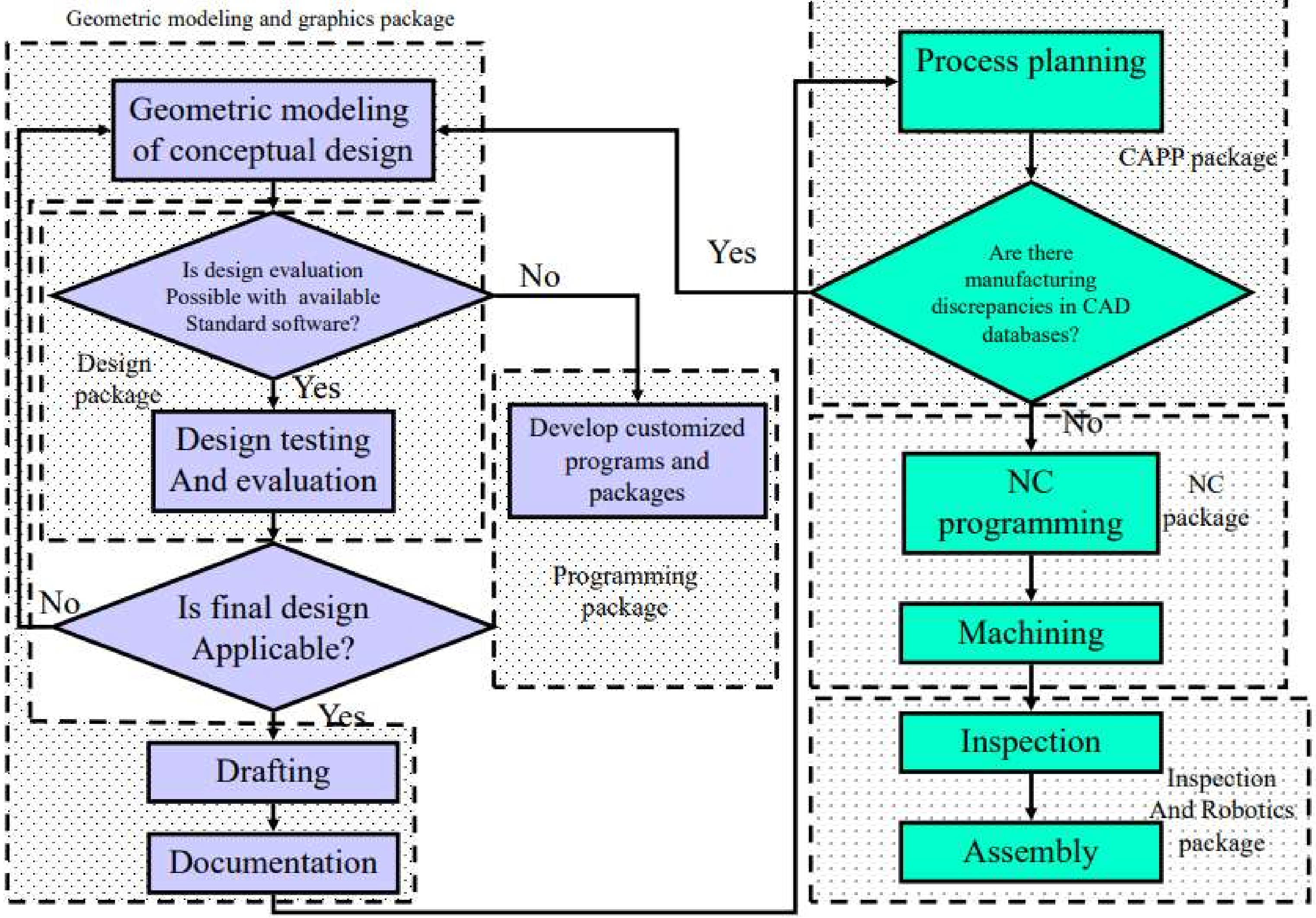


Applications and Analysis of CAD in Product Life Cycle: CAD/CAM/CNC/CAE, and data transfer

- Typical definitions of CAD tools:



•Typical Utilisation of CAD & CAM in industry



Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Poor ergonimic design can make anyone angry, so much so that an afliction has been coined, “wrap rage”. Prodict packaging which is excessive, inappropriate of excessive is often the very first part of the customer experience. Indeed I can recall how – howing worked in hospitals where one regularly comes into contact with disabled, elderly or visually impared patients and visitors to the hospital has instilled me with a lot of 'life experience' and indeed it later led me to write this article, “Nightmare Products which have somehow lasted forever!”.
- http://dfdn.info/personal/nightmare-products.html
- And the most frustrating fact here is that, with a littie vision and foresight at the design stage, many of not all of these issues could have been avoided. So why do so many examples of terrible design persist when they suffer from glaringly obvious problems?
- Here is a list of considerations...

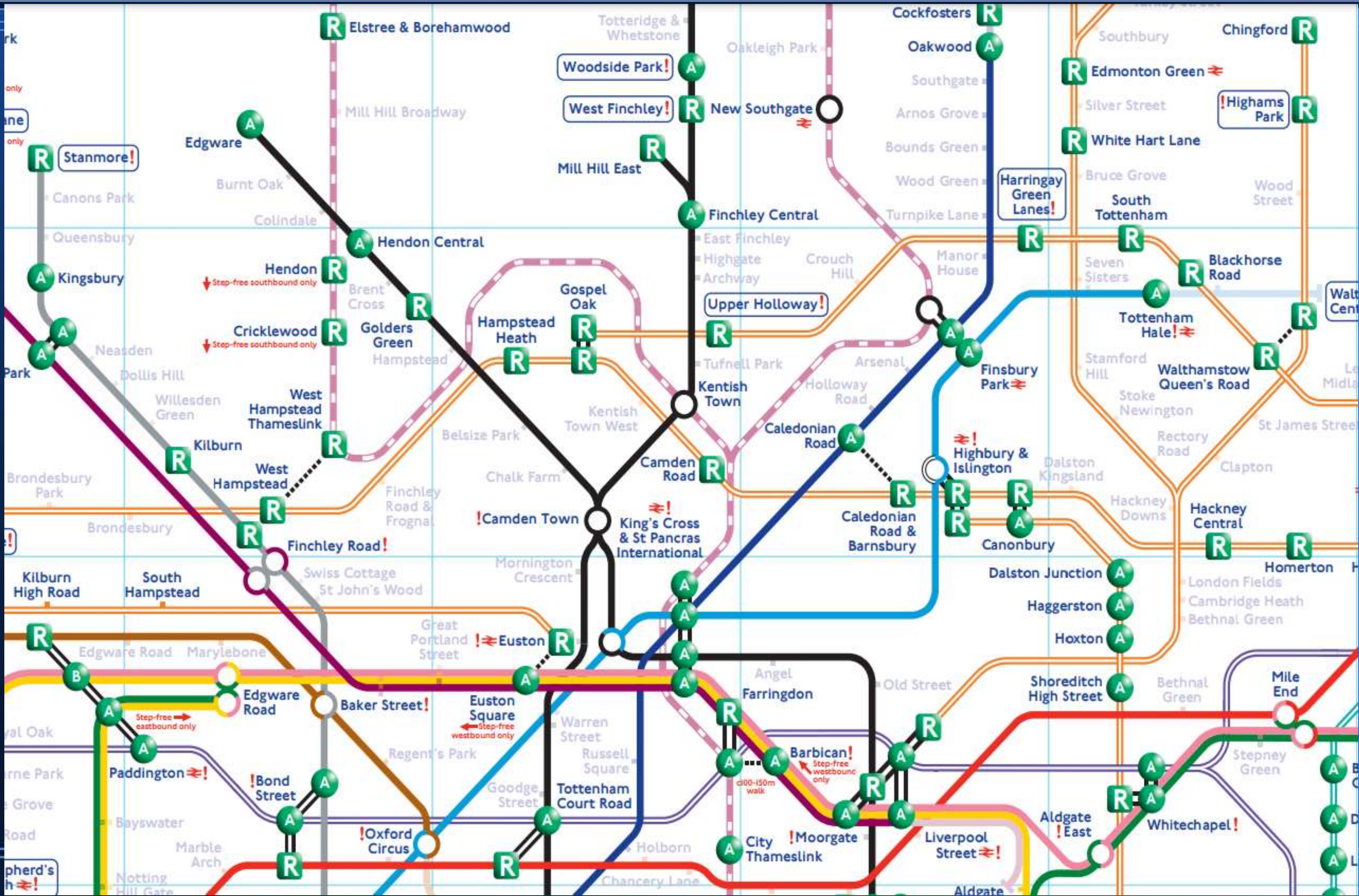
Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- - Lived experience user group accomodation and facilitation (pan-disability focus - physical, sensory, cognitive, multiple and hidden/non-visible impairments), or external, independent engagement with your accessibility working group.
- - Design support relating to your accessibility initiatives. Previous projects I’ve examined included inclusive museum exhibit design, accessible lounge/facility development, and guest services desk design for both visitors and staff members with accessibility requirements.
- - Following an audit of accessibility requirements, creation of an aspirational Inclusive Design Standard or policy for consistent, aspirational and empathetic use by designers and contractors should be agreed upon and implemented, after seeking feedback from people with a wide variety of physical and non-physical impairments.
- - Disability awareness and communication training for guest service members and other operational, front-facing staff should be provided – including to designers!
- - An audit of an organisation's internal culture and inclusive policies, and support in improving inclusive recruitment policies and actions, operational strategies, and the provision of reasonable adjustments for individuals or groups with additional requirements, prior to the design stage, should be conducted.

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- The one thing to commit to consider is the medical model versus social model of disability. This creates a real shift in your mindset. The medical model says, ‘this person can’t do this task because they have a medical condition’. The social model flips this and says, ‘this person can’t do this task because of the way society has built their surrounding environment’.
- Doors are a good example. If the handle is too high, wheelchair users can’t go through them - but it’s not because they can’t use doors! If the doors were automatic, they could use them easlily. We have in fact constructed a world that simply does not work for them”!
- London underground is a very good example. The system was built in the victorian eara taking no account of disability at all! Of course inclusion as a concept had not as yet developed – as evidenced by how few stations have “step free access” - see below.
- https://content.tfl.gov.uk/step-free-tube-guide-map.pdf
- COVID-19 intensified the digital divide. And with a lot of the pandemic’s digital trends staying with us, as we return to a new normal, achieving true digital inclusion in addition to physical inclusion is even more important.

A section of the London Underground map – most stations do not have a green symbol!



Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Inclusive design starts with challenging your assumptions...
- I read about a project to support people with low digital literacy and assumed this would involve a step-by-step guide. But when I spoke to them, the reality was very different. No-one had ever explained to them what 'digital' actually means! Phrases and boxes on a screen meant nothing to them. You don't gain these insights unless you speak directly to people!
- For all projects, this is one of the most important steps in the entire design process – engaging and speaking to all users right from the start!
- This approach saves so much time and costly reworking. Inclusive design should be seamless and appear totally intentional." It should be incorporated into a product 'at the drawing board' not simply 'tacked on afterwards'!

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Inclusive design is about looking at things differently, from the point of view of the end user. -taking a new perspective helps drive innovation.
- The origin of the typewriter is a very good example. The typewriter was invented by Pellegrino Turri for a lover who was blind! At a time when the only means of long-distance communication was through letters, personal thoughts would have to be dictated to someone else because blind and partially sighted people could not write with pen and ink. So, to maintain their privacy and enable freer communication, Turri invented the typewriter!
- It went on to inspire computer keyboards, laptops and smartphones
- By designing for someone who was blind, Turri impacted communications forever.
- When businesses only think about inclusivity from a legal perspective and tick things off a list, they're in danger of producing experiences that don't truly unite – as well as missing out on countless opportunities to innovate. While checklists can provide a foundation; a point of reference, it's important to try and truly understand how typically excluded or underrepresented people may be able interact with a particular product or service (or not!).
-

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- The significance of inclusive design lies in its ability to improve the user experience for a varied audience. About 15% of the world's population faces some form of disability. This is approximately one billion people. Inclusive design entails empathy towards a diverse audience, which is a vital aspect. It enables the creation of an experience that provides a sense of belonging, rather than making users feel excluded.
- A key aim of CAD designers is to simplify the usage of digital products as much as possible. They achieve this by creating personas, testing designs with users, analysing use cases, and incorporating best practices. Yet, designers sometimes overlook the needs of all potential users. Today's design world has a growing emphasis on inclusive design.

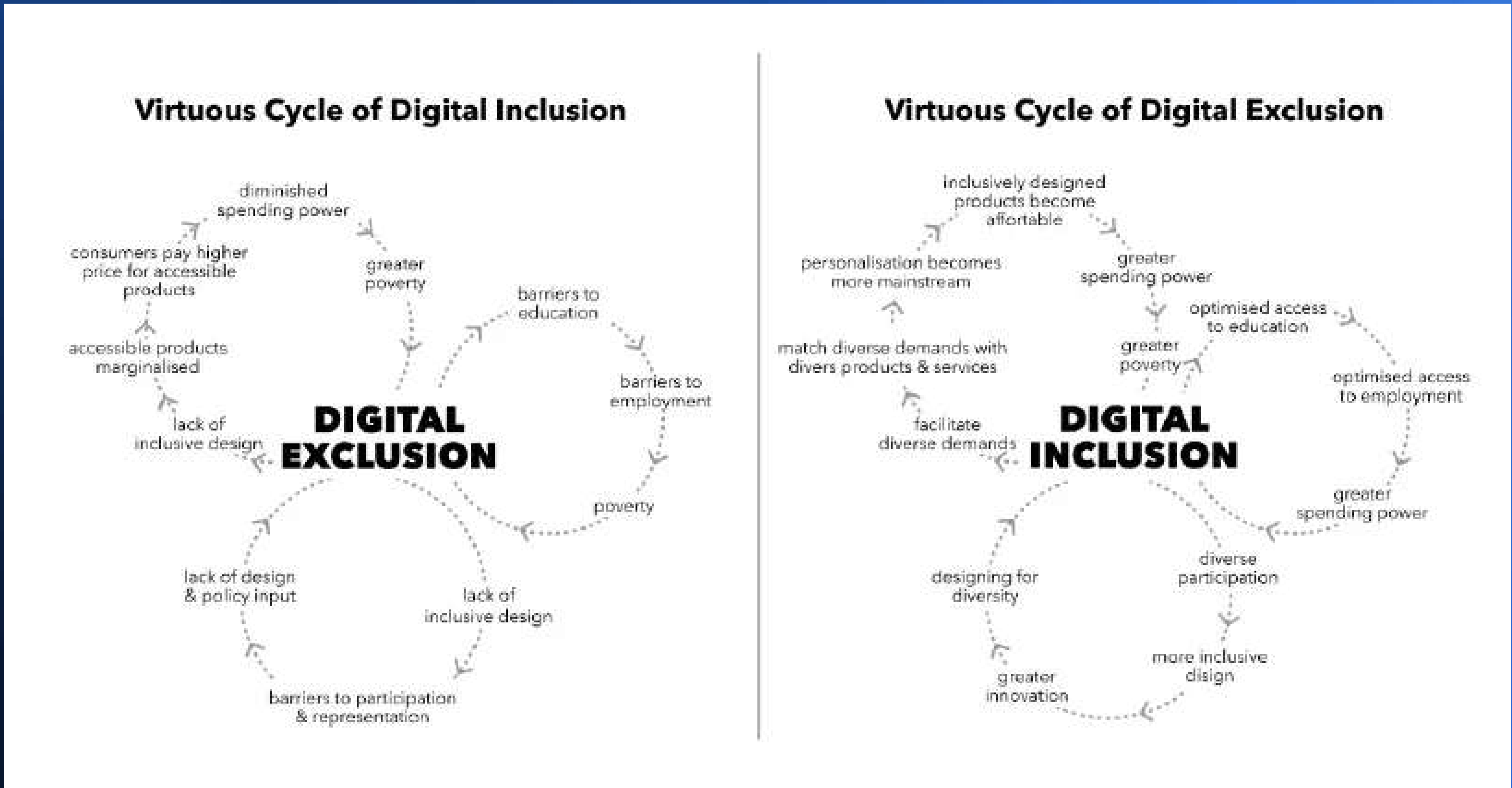
Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- The significance of inclusive design lies in its ability to improve the user experience for a varied audience. About 15% of the world's population faces some form of disability. This is approximately one billion people. Inclusive design entails empathy towards a diverse audience, which is a vital aspect. It enables the creation of an experience that provides a sense of belonging, rather than making users feel excluded.
- So, what is inclusive design?
- To put it simply, inclusive design involves considering the needs of users that belong to a minority group or face oppression. They are often excluded in various aspects of their daily lives. Failure to do so results in exclusion and mismatch how inclusion shapes design.

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Social inclusion focuses on designing for human diversity and equality. That's why design inclusivity is particularly relevant in 2023. There is a heightened emphasis on addressing accessibility and exclusion. The necessity for empathetic approaches is greater than ever.
- Implementing inclusive design principles can assist companies to seem more socially responsible. They promote greater diversity and provide equal opportunities for customers and employees. Inclusivity is an essential aspect of a company's corporate social responsibility efforts. It enhances the firm's public image. Additionally, companies with a more diverse workforce tend to perform better financially.
- The principle of inclusive design extends as much to the virtual world as it does to the physical world.
- And the principles in the next slide apply equally as much to either...

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods



Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Equitable Application: The design should be accessible and appealing to people with a wide range of abilities and should not exclude anyone. Focus on users' privacy, safety, and security when you create inclusive design products and environments.
- To achieve inclusivity, designers can implement features such as high contrast in digital designs. This benefits users with color blindness.
- User Adaptability: Apart from audio, a video should provide closed-captioned subtitles for users who prefer to read instead of listening. This feature is vital for deaf users, but it also accommodates the choices of non-deaf users.
- Straightforward and Intuitive Performance: When users visit a streaming service's website or app, they should be able to find the video they want quickly and easily. This could involve making the most popular titles available on the homepage or providing a large search button for users seeking more obscure content.

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Perceptible Information: In digital designs, textual information should be presented in a clear and concise manner. Place the most critical details at the top. Use bullet points or other formatting tools to further divide the information. We also advise to use images for points and illustrate the written information.
- Error Tolerance: Sometimes users mistakenly tap the “Buy” button on a mobile site. An overlay can appear to confirm the action and provide the option to remove the item.
- Little Physical Effort: Anchor pertinent navigation at the top of a webpage. This way, users can move to different parts of the website without scrolling to the top.
- size and space for approach and use:f or instance, designs must be tailored to the screen size of the device. Buttons must be large enough for users to click them. Not too small so users overlook them on a computer screen and not too large so they take up too much space on a mobile screen.

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- Challenges of the Inclusive Design
- Failure to Identify Exclusion:
- Designers often struggle to identify design bias that indirectly leads to exclusion. Particularly in situations where users face impairments or limitations. These issues can quickly escalate and restrict designers’ ability to create inclusive designs for a diverse audience. Create your own inclusion design lab to acknowledge temporary and situational disabilities. This enables designers to consider alternative scenarios and design for a wider range of user experiences.
- Lack of Diversity:
- To design elements that work in various environments, empathy is crucial. Lack of empathy leads to exclusion and frustration, as seen in issues with gender identity in information forms. Cultural and gender diversity should be considered to create a uniform and empathetic UX design. In 2023, gender diversity will be an essential factor in the inclusive and accessible UX design process.
- Limited Scope:To ensure accessible UX design, consider different disabilities and limitations. Use informative images with ALT text, provide suitable descriptions or captions for graphics, and avoid embedding text in images. Ensure enough contrast between text and color, using tools to achieve the correct ratio for higher visibility.

Case Study and Demonstration of inclusivity, diversity, and inclusion in Design Methods

- When designing products, it’s common to assume that if it works for us, it’ll work for everyone else. Yet, this is not true!!! Product developers are not a representative sample of the target user population. They often have different abilities.
- Creating products that are inclusive assists teams in understanding and sharing the feelings of their users. This eliminates partiality from the process of designing products.
- Inclusive product design can also eradicate any obstacles during a user’s interaction with the product. This can result in a smaller number of users who drop using the product and a bigger number of users who adopt it. This can lead to more successful product outcomes over time.
- Moreover, incorporating all users, particularly those who have been traditionally marginalised, benefits business:
- enhances their devotion to the product,
- enhances user retention,
- and bolsters your brand’s reputation.
- So let’s take a look at some of the best practices for inclusive design in product development!

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

- Do not Consider Accessibility a Secondary Priority!!!Teams often ignore inclusivity and accessibility until the product is complete. But this approach only checks the accessibility boxes and may not create a genuinely inclusive product.Also, implementing adaptations at this stage may be expensive and complicated. Incorporating inclusivity from the start is more cost-effective and straightforward. It may be incorporated into customer research to address accessibility requirements alongside functionality needs.
- Incorporate Ongoing Discovery as a Component of the Process:
- Since user requirements are continually changing, the product discovery process is not a single event.
- Business consistently seeks out new user concerns, requirements, and strategies to tackle them. We recommend to keep an eye out for chances to enhance the accessibility and inclusivity of your product.

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

- Inclusive Design = Less Assumptions, More Testing!
- When developing a product, we typically hold certain beliefs about our users and how the product fulfills their requirements. While some of these beliefs may be correct, others could be entirely mistaken. The only way to verify their accuracy is by conducting tests with actual users.
- Beta testing:
- During the building stage of a product, beta testing can be an effective method to assess its design. This involves releasing the product to a group of users and observing their actions to identify any accessibility problems. If users encounter difficulties or fail to complete certain tasks, this may show issues you need to address.
- Based on the findings, designers make modifications to the sketch, and carry out further testing to evaluate whether changes result in improved conversion rates. The success of beta testing depends on selecting the appropriate testers. They should represent a diverse range of individuals that are part of the target audience.

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

- Usability testing interviews:
- To enhance the inclusive design of your product, conduct user interviews after its launch to assess its usability. Like beta testing, it’s essential to involve a diverse group of users in the interviews. To ensure enough participation, you may need incentives such as vouchers or discount codes. There are various techniques for usability testing, including the five-second test. In this technique, you show the product to users for five seconds, then interview them about their impressions. In addition to interviews, there are other methods like first-click testing, eye-tracking, and session replays.
- Continuous user feedback collection:
- While your users are actively using your product, take advantage of in-app surveys to gather feedback on their user experience. These surveys are user-friendly to design and can be triggered to target specific user segments.
- To enhance inclusivity and accessibility, it’s crucial to obtain qualitative feedback. You may achieve it through open-ended questions included in the surveys. Don’t forget to incorporate this element into your survey to ensure valuable feedback from users.

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

- Consider Cultural Barriers and Use Localisation:
- Product localisation creates an inclusive product experience for users with different language and cultural backgrounds. It goes beyond just translating UI copy or content. It may involve adapting the UI for various scripts and cultural differences to make it feel like it was originally designed for them.
- In product localisation, you shall consider all touchpoints in the user journey where they interact with the product. These touchpoints include:
- 1. Display ads, social media posts, and search keywords;
- 2. Landing and product pages;
- 3. Sign-up process, onboarding steps;
- 4. Main UI, modals, and account management;
- 5. Onboarding emails, use case emails, missed steps emails, newsletters;
- 6. Help section or customer support;

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

- 7. Upsell and demo emails;
- 8. Webinars, tutorials, demos, walkthroughs;
- 9. Pricing page, checkout page, confirmation emails;
- 10. Engagement emails, win-back emails, feedback emails;
- 11. Renewal emails, notification emails.

How to Integrate Inclusive Design into your Product Development Process? The all-important stages...

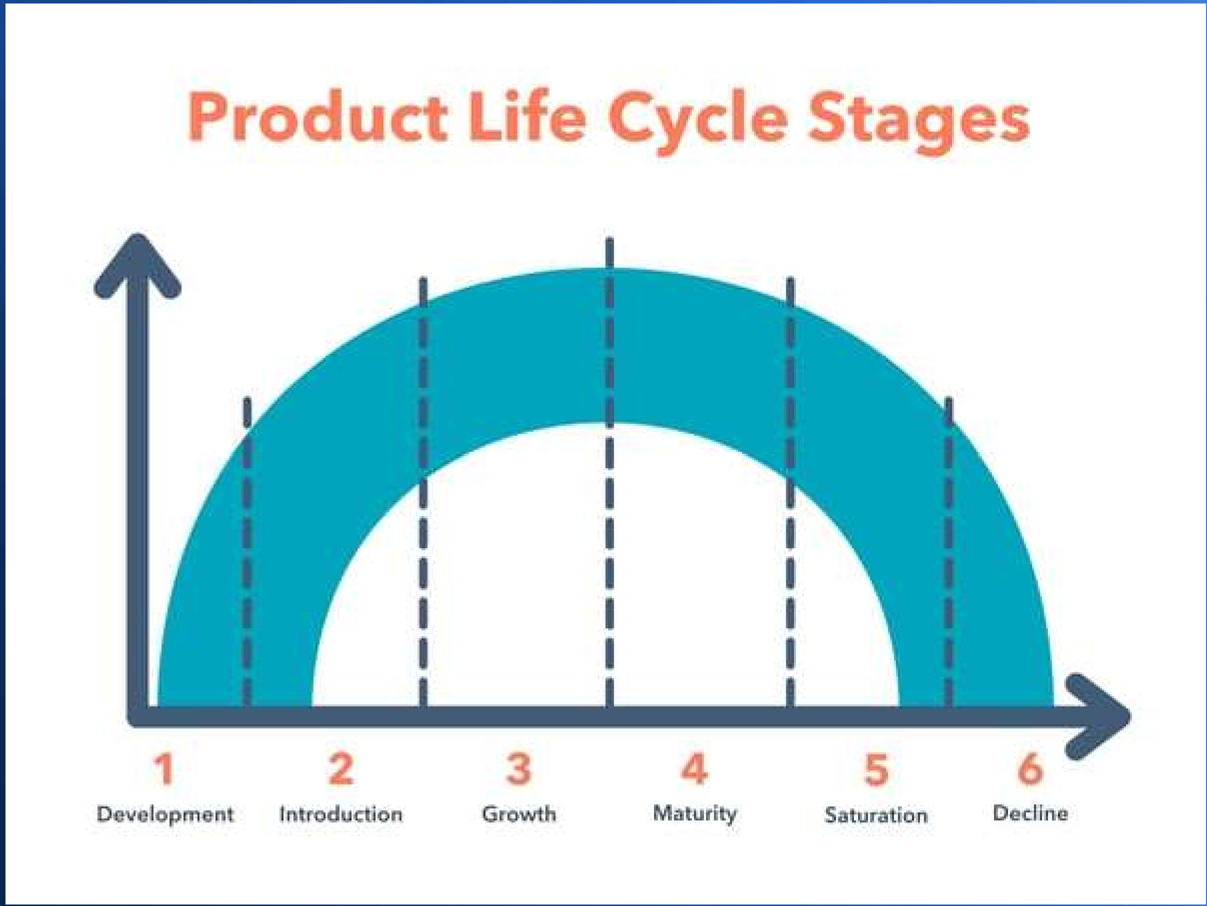
- Build a Diverse Team...
- To create an inclusive and accessible product, involve individuals with diverse perspectives.
- Build a product team with members from different cultures and backgrounds. This can provide unique perspectives and help to identify issues and opportunities to create an exceptional product.
- Foster collaboration:
- To create durable product experiences, it is not enough to depend entirely on the product team's skills. Thus, it is important to ensure that the product team collaborates closely with other teams: UI and UX designers, marketing colleagues, and customer-facing staff.
- These individuals can provide distinctive viewpoints and understandings regarding user issues. They know how to improve the inclusivity and accessibility of your product.

Address challenges of information security in product lifecycle management

- The 6 Stages of the Product Life Cycle
- Firstly we need to talk about the product lifecycle itself...
- What is the product life cycle?
- The product life cycle is the succession of stages that a product goes through during its existence, starting from development and ultimately ending in decline. Business owners and marketers use the product life cycle to make important decisions and strategies on advertising budgets, product prices, and packaging.
- In the marketing industry, the typical depiction of the product life cycle only has four main stages — Introduction, Growth, Maturity, and Decline. Tthese are vital for a product, but the two stages “Development” and “Decline” aren’t nearly covered enough...

Address challenges of information security in product lifecycle management

- A more complete cycle...



Address challenges of information security in product lifecycle management

- As designers – or marketers - it's important to understand how your tactics and strategies change depending on the stage your product is in. Let’s explain each of the six stages of the product life cycle.
- **1. Development**
- The development stage of the product life cycle is the research phase before a product is introduced to the marketplace. This is when companies bring in investors, develop prototypes, test product effectiveness, and strategiss their launch.
- In this stage, companies typically spend a lot of money without bringing in any revenue because the product isn't being sold yet.

Address challenges of information security in product lifecycle management

- This phase can last for a long time, depending on the complexity of the product, how new it is, and the competition. For a completely new product, the development stage is particularly difficult because the first pioneer of a product isn't always as successful as later iterations.
- **2. Introduction**
- The introduction stage happens when a product is launched in the marketplace. This is when marketing teams begin building product awareness and targeting potential customers. Typically, when a product is introduced, sales are low and demand builds slowly.
- In this phase, marketers focus on advertising and marketing campaigns. They also work on testing distribution channels and building product and brand awareness.

Address challenges of information security in product lifecycle management

- Introduction Stage Marketing Strategy:
- This is where the fun begins. Now that the product is launched, you can actually promote it using inbound marketing and content marketing.
- Education is vital in this stage. If your marketing strategies are successful, the product goes into the next stage — growth!
- **3. Growth**
- During the growth stage, consumers have accepted the product in the market and customers are beginning to truly buy in. That means demand and profits are growing, hopefully at a steadily rapid pace.
- The growth stage is when the market for the product is expanding and competition begins developing. Potential competitors will see your success and will want in.
- Growth Stage Marketing Strategy: During this phase, marketing campaigns often shift from getting customers' buy-in to establishing a brand presence so consumers choose them over developing competitors. Additionally, as companies grow, they'll begin to open new distribution channels and add more features and support services. In your strategy, you'll advertise these as well.

Address challenges of information security in product lifecycle management

- **4. Maturity**
- The maturity stage is when the sales begin to level off from the rapid growth period. At this point, companies begin to reduce their prices so they can stay competitive amongst the growing competition.
- This is the phase where a company begins to become more efficient and learns from the mistakes made in the introduction and growth stages. Marketing campaigns are typically focused on differentiation rather than awareness. This means that product features might be enhanced, prices might be lowered, and distribution becomes more intensive.
- During the maturity stage, products begin to enter the most profitable stage. The cost of production declines while the sales are increasing.

Address challenges of information security in product lifecycle management

- Maturity Stage Marketing Strategy
- When your product has become a mature offering, you may feel like you're "sailing by" because sales are steady and the product has been established. But this is where it's critical to establish yourself as a leader and differentiate your brand.
- Continuously improve upon the product as adoption grows, and let consumers know in your marketing strategy that the product they love is better than it was before. This will protect you during the next stage — saturation.
- **5. Saturation**
- During the product saturation stage, competitors have begun to take a portion of the market and products will experience neither growth nor decline in sales.
- Typically, this is the point when most consumers are using a product, but there are many competing companies. At this point, you want your product to become the brand preference so you don't enter the decline stage.

Address challenges of information security in product lifecycle management

- Saturation Stage Marketing Strategy
- When the market has become saturated, you'll need to focus on differentiation in features, brand awareness, price, and customer service. Competition is highest at this stage, so it's critical to leave no doubt regarding the superiority of your product.
- If innovation at the product level isn't possible (because the product only needs minor tweaks at this point), then invest in your customer service and use customer testimonials in your marketing.

Address challenges of information security in product lifecycle management

- **6. Decline**
- Unfortunately, if your product doesn't become the preferred brand in a marketplace, you'll typically experience a decline. Sales will decrease during the heightened competition, which is hard to overcome.
- Additionally, new trends emerge as time goes on, just like the CD example I mentioned earlier. If a company is at this stage, it'll either discontinue its product, sell the company, or innovate and iterate on its product in some way.
- Decline Stage Marketing Strategy
- While companies would want to avoid the decline stage(!), sometimes there’s no helping it — especially if the entire market reached a decline. In your marketing strategy, you can emphasize the superiority of your solution to successfully get out of this stage.

Address challenges of information security in product lifecycle management

- To extend the product life cycle, successful companies can also implement new advertising strategies, reduce prices, add new features to increase their value proposition, explore new markets, or adjust brand packaging.
- The best companies will usually have products at several points in the product life cycle at any given time. Some companies look to other countries to begin the cycle anew.
- **Importance of the Product Life Cycle**
- The product life cycle is important because it informs an organisation's management and decision-makers how well a product is performing and what strategic actions it will take to succeed. This helps companies allocate resources like staff, budgets, shows which products should be prioritised, and where the company should innovate next.

Address challenges of information security in product lifecycle management

- Other benefits of using the product life cycle include:
- Make better marketing investments and decisions;
- Easier to make long-term plans;
- Allows for better decision making with accurate information on performance;
- Easier to streamline current processes within your company.
- **Product Life Cycle Limitations:**
- While using the PLC method certainly helps stakeholders plan, it does have limitations. The cycle breaks down performance over several stages, but unfortunately there is no way to tell how long each stage will last.
- Complicating things further, not all products will move through these stages at the same pace. For example, a product may take longer to decline than others. Plus product managers run the risk of not dedicating enough effort and resources into a particular product if they think the product will decline, creating planned obsolescence – even if customers still use it.

Two Product Life Cycle Examples...

- 1. The Typewriter
- The typewriter was the first mechanical writing tool — a worthy successor to pen and paper! Ultimately, however, other technologies gained traction and replaced it.
- Development: Before the first commercial typewriter was introduced to the market, the overall idea had been developed for centuries, beginning in 1575.
- Introduction: In the late 1800s, the first commercial typewriters were introduced.
- Growth: The typewriter quickly became an indispensable tool for all forms of writing, becoming widely used in offices, businesses, and private homes.
- Maturity: Typewriters were in the maturity phase for nearly 80 years, because this was the preferred product for typing communications up until the 1980s.

Two Product Life Cycle Examples...

- Saturation: During the saturation stage, typewriters began to face fierce competition with computers in the 1990s.
- Decline: Overall, the typewriter couldn't withstand the competition of new emerging technologies, and eventually the product was discontinued.
- **2. Cable TV**
- Remember the days of switching TV channels to find what to watch? I do — and they feel distinctly like something of the past! While cable TV is still around for now, it's safe to say that it's nearing the decline stage.
- Development: Cable TV was developed in the first half of the twentieth century. John Walson has been credited with its invention.
- Introduction: The first commercial television system was introduced in 1950, and by 1962, the technology saw the first hints of growth.

Address challenges of information security in product lifecycle management

- Growth: After a decades-long freeze on cable TV's development (due to regulatory restrictions in some countries), the technology began gaining traction, and by 1980, more than 15 million households in the USA had cable. But it did not arrive in the UK until the 1990s.
- Maturity: Cable TV matured around the 1990s. Around seven in ten households had cable.
- Saturation: The start of the 21st century saw an oversaturation of this technology, and it also started to compete with other modern developments such as on-demand services and high-definition TV (HDTV). While the internet was still in its nascent stages, it would soon gain on cable TV as well.
- Decline: From 2015 onwards, cable TV experienced a marked decline. Online video streaming services such as Netflix and Hulu have taken precedence — and this trend is set to continue.

International Product Life Cycle

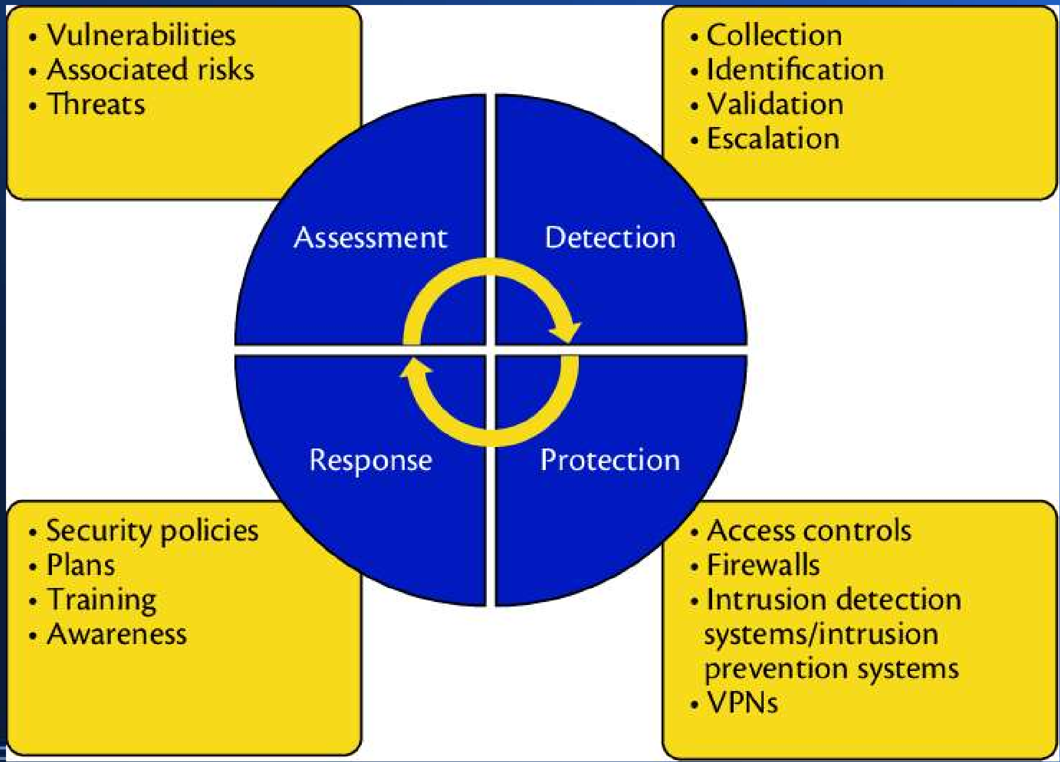
- Not all products need to face the decline stage. Companies can extend the product life cycle with new iterations and stay afloat as long as they have several products at various points of the product life cycle.
- he international product life cycle (IPL) is the cycle a product goes through in international markets. As products begin to mature and companies want to avoid the decline stage, they'll typically begin to explore new markets globally.
- When products reach mass production, manufacturing and production shift to other countries as well.
- The international product life cycle stages are identical to that of a normal product life cycle. The development stage looks different, however, because local customs and regulations can affect how long it takes to bring the product to a new marketplace.

Address challenges of information security in product lifecycle management

- However, once you lay the groundwork in a new marketplace, your competitors will be sure to follow, and the life cycle stages will continue up until saturation and eventually decline. Your option is to either expand into another market or learn from prior mistakes and innovate before the decline stage rolls around.
- But what are the implications for information security? In a world where much development involves firmware and networking, *the security implications of an interconnected online world need to be addressed in advance of a product's development, not after it gets hacked!*

Address challenges of information security in product lifecycle management

- The model presented here follows the basic steps of IDENTIFY – ASSESS – PROTECT – MONITOR. This lifecycle provides a good foundation for any security program. Using this lifecycle model provides you with a guide to ensure that security is continually being improved.



Address challenges of information security in product lifecycle management

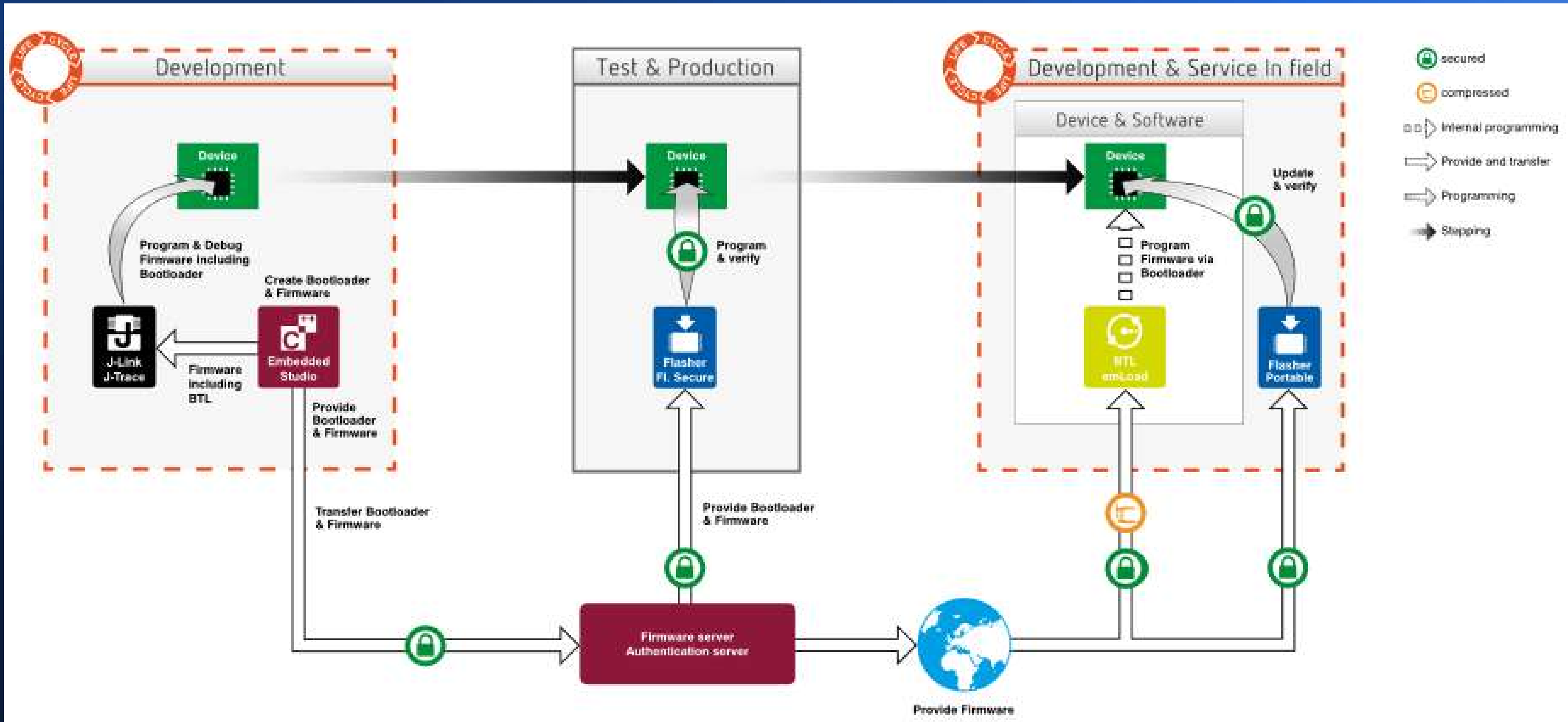
- The 3 Stages in Secure Product Lifecycle Management – a case study
- Nearly every embedded software developer working in the IoT (internet of things) Industry is now building secure devices. Developers have been mostly focused on how to handle secure applications and the basic microcontroller technologies such as how to use Arms TrustZone or leverage multicore processors. A looming problem that many companies and teams are overlooking is that figuring out how to develop secure applications is just the first step. There are three stages to secure product lifecycle management and in today’s post, we will review what is involved in each stage.
- In a world where everything from industrial processes to domesti shopping, from banking to product development involves networking, we need to think about the implications for product development...

Address challenges of information security in product lifecycle management

- As a quick overview, the stages, which can be seen in the illustration are:
- Development
- Test and Production Deployment
- Maintenance and In-field Servicing
- Let us look at each of these stages in a little more depth.

Address challenges of information security in product lifecycle management

- The Secure Product Lifecycle Management cycle



Address challenges of information security in product lifecycle management

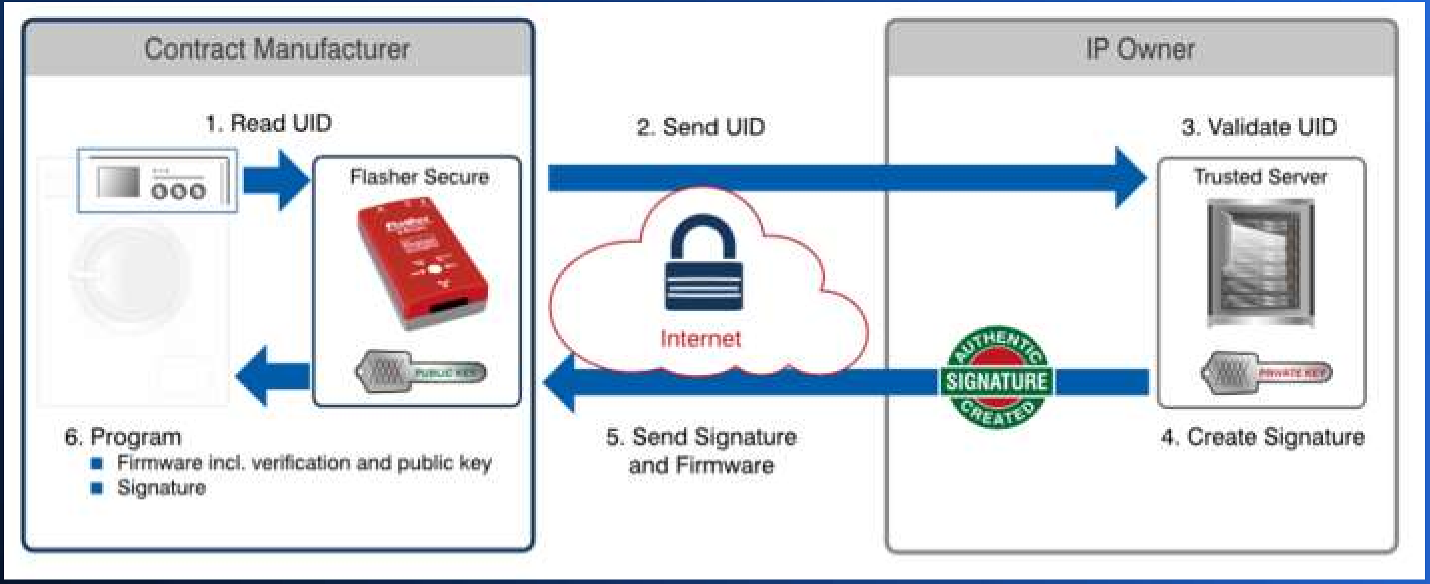
- **Stage 1 – Development**
- Development is probably the area that most developers are the most familiar with, but at the same time, the area that they are learning to adapt to the most! Many developers have designed and built systems without ever having to take security into account. Development involves a lot more than just deciding which components to isolate and how to separate the software into secure and non-secure regions.
- For example, during the development phase developers now need to learn how to develop in the environment where a secure bootloader is in place. They need to consider how to handle firmware fallbacks, if they are allowed and if so, under what conditions. Firmware images may need to be compressed on top of the need for authentication.
- While the development stage has become more complicated as a consequence, developers should not struggle too much to extrapolate their past experiences to developing secure firmware successfully.

Address challenges of information security in product lifecycle management

- **Stage 2 – Test and Production Deployment**
- The area that developers will probably struggle with the most is the test and production deployment stage. Testing secure software requires additional steps to be taken that authenticate debug hardware so that the developer can access secure memory regions to test their code and successfully debug it. Even more importantly, care must be taken to install that secure software onto a product during production.
- There are several ways this can be done, but one method is to use a secure flashing device like SEGGERS Flasher Secure. These devices can follow a multistep process that involves validating a user ID which allows the secure firmware to be installed on the device. The devices themselves limit how many and on what devices the firmware can be installed which helps to protect a team's intellectual property and prevents unauthorised production of a product.

Address challenges of information security in product lifecycle management

- Stage 3 – Maintenance and In-field Servicing:
- Finally, there is the maintenance and in-field servicing stage which is a partial continuation of the development phase. Once a product has been deployed into the field, it needs to be securely updated.



Address challenges of information security in product lifecycle management

- Updates can be done manually in-field, or they can be done using an over-the-air update process. This involves a device being able to contact a secure firmware server that can compress and encrypt the image and transport it to the device. Once the device has received the image, it must decrypt, decompress and validate the contents of the image. If everything looks good, the image can then be loaded as the primary firmware for the device.
- Conclusions:
- There is much more to designing and deploying a secure device than simply developing a secure application. The entire process is broken up into three main stages that we have looked at in greater detail today. Keep in mind the principle that any product development should address security issues from the drawing board, 'tacking it on later' is not good enough! Ultimately, we have only just scratched the surface!

The Future of ECAD

- The global Electrical Computer-Aided Design (ECAD) market size was valued at USD 2190.0 million in 2021 and is expected to expand at a CAGR of 8.84% during the forecast period, reaching USD 3640.0 million by 2027.
- With this growth projection, it is clear that new ECAD software is under development and we will see a lot of inovation in the field.
- I hope you have found this course interesting and – of course, I wish you all best of luck in the exam! We will look at some revision techniques to help you on the way.
- David

Exam Revision Techniques

- Take some time to understand your learning style
- When it comes to finding the best revision techniques for students, it all begins with understanding how you learn best, e.g. what your learning style is. There are lots of different learning styles out there, with many turning to the VARK theory to understand their preferred learning style. In essence, the VARK theory identifies us as being one of the following learners: visual, aural, read (or write), or kinaesthetic – take the test below to find out which type of student you are!
- Once you know the method of learning that suits you best, simply tailor each of your revision sessions by choosing the techniques that will make remembering the information much easier for you. You'll find that your revision becomes far easier, engaging, and effective on the whole.
- https://vark-learn.com/the-vark-questionnaire/

Exam Revision Techniques

- Organise your notes ahead of time
- To ensure you can kick-start your revision in the most efficient way possible, it's a good idea to (if they aren't already) organise, label, and clearly order your subject notes so that they are easy to read through and use as part of the revision process.
- When you sit down on your first few days of revision, the last thing you want to have to do is waste time finding and filing all your class notes together for you to then begin your revision. Taking the time outside of class to condense and organise your notes into a formulated system will have endless benefits, both at helping you to reconfirm your understanding of the content after class, but also making your revision far more manageable.

Exam Revision Techniques

- Use mind maps to connect ideas
- When it comes to your revision, do you find yourself struggling with remembering lots of new information? Or understanding how different topics relate to each other? Well, mind maps may be key to helping you succeed!
- In essence, the theory behind using mind maps is that making associations between related ideas can help us to memorise information quicker and faster – making it a very effective revision technique.
- Mind maps begin with one central theme or topic. From here, you can then create branches from this central idea with other related ideas that you want to develop or visualise. From these branches, you can add further detail and information, with keywords helping you to summarise information, include key terminology, and visually connect ideas between one another.
- Having a topic summarised into a mind map on one big sheet of A3 paper can be hugely beneficial to information retention, especially if you also use visual aids to help summarise processes or definitions.

Exam Revision Techniques

- Complete as many past papers as possible!
- Another highly effective revision technique to help you prepare for your exams is to get familiar with past papers. After all, there's no point learning all that content if you don't know how to apply it to the exams.
- Past papers can be great at helping you become familiar with the format of exams, including the different types of question styles and time restraints. Then, when it comes to the real thing, you'll know exactly what to expect.
- But aside from this, completing past pacers can also be a good way to test your current understanding of a subject and identify any gaps of knowledge or areas that you're struggling with.

Exam Revision Techniques

- Lastly, mix your study habits up to keep it engaging
- For some ideas on how to keep your revision engaging, try using one or some of the following techniques:
- Watch video demonstrations or documentaries
- Listen to podcasts
- Organise a group study session
- Mix your study time between at-home and at a library or local café
- Write about your topic as if you were telling a story
- Try teaching a topic to a friend or family member who has little to no knowledge of it
- And finally, do some revision with other members of the class!

Important notice!
重要通知！

When I taught the previous Engineering course in May, the results were delayed. When you take the exam, please ensure that you clearly mark your English name, Chinese name and student number on the exam paper.

This will expedite marking (and hence results!) for all.

Many thanks.

当我在五月份教授之前的工程课程时，结果延迟了。 参加考试时，请确保在试卷上清楚地标明自己的英文姓名、中文姓名和学号。

这将加快所有人的标记（以及结果！）。

非常感谢。

New information

- Any new announcements which I become aware of during the progress of the course will be published here.

New information