

CHAPTER 2

THEORETICAL FOUNDATION

2.1 Skateboarding

Skateboarding is a kind of sport which describe as the act of riding on or performing tricks with a skateboard. A person who skateboards is referred to as a skateboarder or more generally called a skater. Skateboard is a short narrow board having a set of four wheels mounted under it, ridden in a standing or crouching position and often used to perform stunts [2]. Skateboard is said to be not easily define since unlike most other sports, there is no governing body which declares any regulations on what constitutes a skateboard or the parts from which it is assembled. Thus, the definition of a skateboard can vary in appearance and performance as well as size.

2.1.1 Skateboarding style

A skateboard style is the preferred way a skater rides a skateboard. There are many skateboarding style categorized not only from the actual riding of the skateboard itself but also clothing, music, etc. There are various styles of riding a

skateboard and each of them has different methods and techniques. In general, there are two styles of skateboarding based on the environmental matter.

2.1.1.1 Street skateboarding

Street skateboarding is the act of riding a skateboard on paved surface whether that surface is found at a public school, a shopping mall, or somewhere else. Street skateboarding, as it is most commonly known among skaters may include skate tricks such as ollie or ollie variations. Basically, the act of street skateboarding requires the skater to utilize objects which are found in urbanized settings, such as curbs, ledges, handrails, stairs, and other obstacles.



Figure 1 Street skateboarding [3].

2.1.1.2 Vertical skateboarding

On the contrary, vertical skateboarding or often called vert skateboarding means skating on ramps and other vertical structures specifically design for skating. For example:

- Half pipe: a U-shaped ramp of any size, usually with a flat section in the middle.
- Vertical ramp: a half pipe, usually at least 8 feet tall, with steep sides that are perfectly vertical near the top.
- Mini ramp: a vertical ramp or half pipe shorter than 8 feet tall.
- Quarter pipe: a half of a half pipe.
- Bowl: a pool like structure used for skating in.



Figure 2 Vertical skateboarding [4].

2.1.2 Skateboard parts

There are several parts of a skateboard besides just the board and the wheels. The different parts of a skateboard are significant to its performance during use. The basic parts of a skateboard are the deck, the trucks, the grip tape, the wheels and the bearings. Explanations of the following skateboard parts are based on the most popular and modern forms of skateboarding.

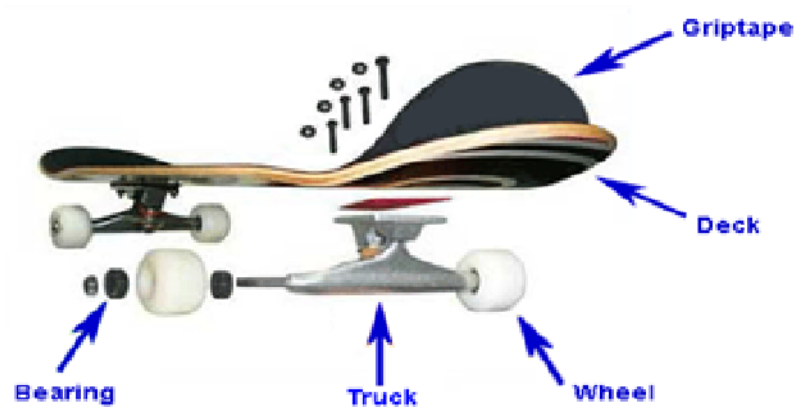


Figure 3 Skateboard parts [5].

2.1.2.1 Deck

The deck is the flat standing surface of a skateboard. It is made by pressuring separate sheets of maple ply together with very strong glue in between them. Sometimes more other exotic materials, such as fiberglass,

bamboo, Kevlar, and plastic are incorporated into deck construction usually to lighten the board or to increase its strength or rigidity. The size of decks can be varied, the smaller boards are easier to control and the larger boards give a more stable feeling. Most modern decks are using 7 to 8½ inches wide. All skateboard decks have a concave, which is where instead of the deck being flat, it curves up slightly. This makes the deck more flexible and stronger so that it will not break easily.

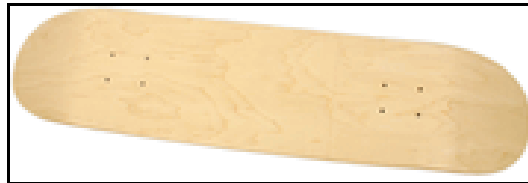


Figure 4 Skateboard deck [6].

2.1.2.2 Grip Tape

Grip tape is the sandpaper like black material attached on the top of the deck with adhesive, used to increase the grip between the deck and the skater's feet. Usually skaters make some patterns into their grip tape to make their boards unique and also to help them simply tell between the nose and the tail of the board.



Figure 5 Grip tape [7].

2.1.2.3 Trucks



Figure 6 Skateboard trucks [6].

Trucks are the parts of the skateboard that attach underneath which connect to the wheels and deck. It provides the turning capabilities for the board and help skaters to turn and lean in a particular direction, for example if skater leans to the right, it turns right. Most trucks are made from an aluminum alloy nowadays and they are very strong. Trucks have many different shapes, sizes, and colors. Skaters can adjust how much the truck

turns by tightening or loosening the trucks. The tighter it is, the less the truck will turn and on the other hand the more loosen then the more they will turn.

The trucks are comprised of smaller parts:

- The hanger – the T-shaped part that holds the axel.
- The axle – runs through the hanger and the wheels are mounted onto each end of the axel.
- The base plate – attaches the truck to the deck.
- The kingpin – large bolt that holds everything together.
- The bushings – rubbery rings that are fit into the truck

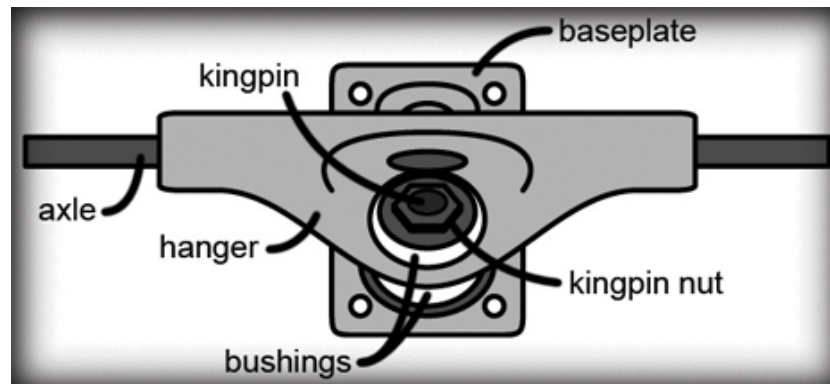


Figure 7 Elements of skateboard trucks [8].

2.1.2.4 Wheels

Skateboard wheels are made out of *polyurethane* and sized between 39 and 70 millimeters in diameter. The wheels affect the smoothness of the

ride on a skateboard and the speed. Hard and large wheels will go faster than soft and small wheels. Their hardness is measured by durometer, a number ranging from 0 to 100. Soft wheels have a durometer of about 85 and hard wheels have a durometer of 98 or higher.



Figure 8 Skateboard wheel [6].

2.1.2.5 Bearings



Figure 9 Skateboard bearings [9].

Basically, skateboard wheels turn freely because of an anti-friction device called a bearing that is inserted between each wheel and axle. The

bearings reduce the rubbing between a moving wheel and a fixed non moving edge. Bearings are the skateboard parts that enable the wheels to turn and have different types. Many skateboard bearings are graded according to the ABEC scale, mostly between ABEC1 and about ABEC7. Basically, the larger the number, the faster the bearings normally go.

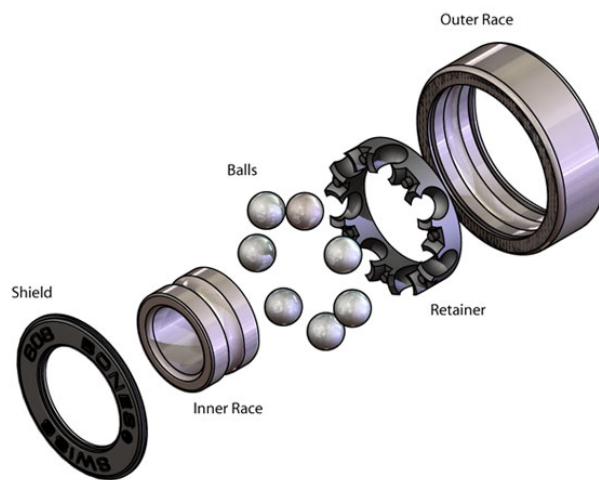


Figure 10 Skateboard bearings elements [10].

2.2 Simulation

According to Institute for Simulation and Training (IST) in University of Central Florida, simulation is a specific application of models to arrive at some outcome. In general, Simulation can be defined as the act of imitating the behavior of some situation or some process by means of something suitably equivalent, especially for the purpose of study. In specific, simulation in computer science can be put in plain words as the

technique of representing the real world by a computer program. Simulation has been used for thousands of years to train, explain, and also entertain. Often it deliberately emphasizes one part of reality at the cost of other parts, sometimes it is done to focus user attention on an important aspect of the simulation.

According to IST, simulations commonly come in three styles. Those styles are live, virtual, and constructive. A simulation also may be a combination of two or more styles.

2.2.1 Live Simulation

It usually involves humans with or without using equipment doing having activities in a setting where they would operate for real. For example, in war games with soldiers out in the field, the time is continuous as in the real world.

2.2.2 Virtual Simulation

It usually involves humans with or without using equipment in a computer-controlled setting. For example, in a flight simulator, the time is in separate steps allowing users to focus on the important factor.

2.2.3 Constructive Simulation

It usually does not involve humans or equipment as contributors. This type of simulation is determined more by the appropriate cause and effect events rather than by time. For example, the estimated path of a hurricane can be constructed

through the accurate calculation of temperatures, pressures, wind currents and other weather conditions.

2.3 Modeling

A model in computer science is a computer-based description and representation of a three dimensional object. 3-Dimensional computer model provide a fundamental resource to several industries, for example:

- Prototyping for product design.
- Architectural visualization.
- Scientific simulation.
- Archeological reconstruction.

Modeling is the act of representing something usually on a smaller scale [11]. In general, modeling can be defined as the process of representing a real world object or phenomenon as a set of mathematical equation. More specific, the term is often used to describe the process of representing 3-dimensional objects in a computer.

There are few terms that are regularly used in modeling process, such as:

- Face

Face is a flat, triangular area of a model. Models are made up of faces, and each face is surrounded by three lines called edges.

- Polygon

Polygon is a flat area of a model, usually rectangular, made up of two or more faces. It is surrounded by edges, usually four but sometimes more.

- Vertex

A vertex is a point where two edges meet. The plural of vertex is *vertices*.

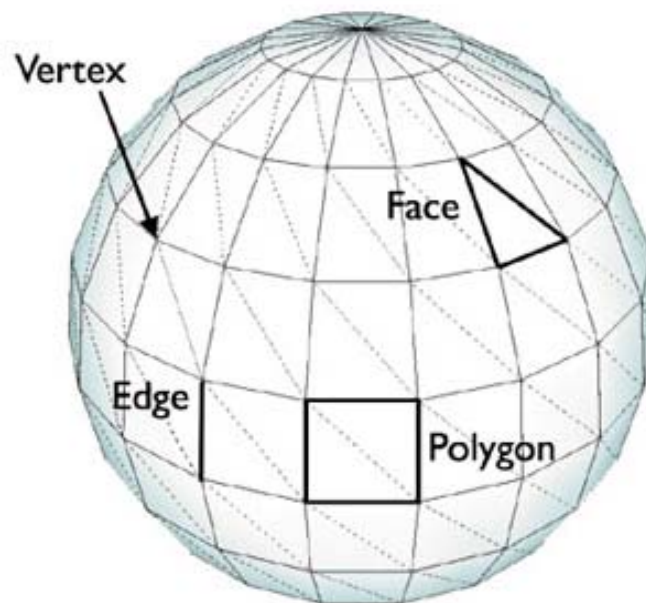


Figure 11 A vertex, face, polygon, and an edge [12].

2.3.1 Box Modeling

Box modeling is a modeling technique that begins with a simple object such as a box or a sphere. Using box modeling, a modeler can model pretty much anything including characters of any kind.

2.3.2 Character Sheet

A character sheet is a guide to human proportions, and it can save 3D modelers a great deal of time in creating characters. It is used as a reference document to help animators draw characters. Designer can just make use of the front and side views of the character shown in the character sheet with any number of expressions, poses and costume details that the designer wants to include.

First of all, designer scan his or her character sheets and then map the scanned images onto polygons in his or her 3D package and work straight over that scanned image in the viewport. By means of the sheet shown in viewport, designer can position the geometry quickly and reduce the time spent change the character model's position and scale.

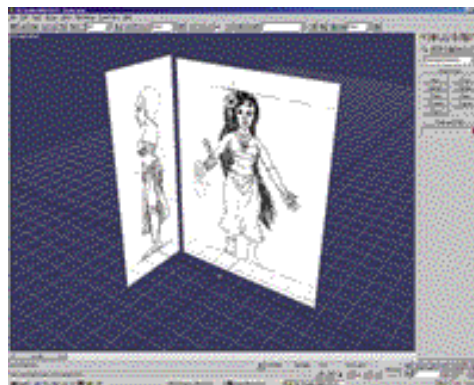


Figure 12 An example of a character sheet [13].

2.3.3 Polygon Modeling / Component Modeling

Polygons are the basic building block for all geometry objects in 3D and can be manipulated in a variety of ways. Their flexibility makes them the geometry of choice for serious modelers. Polygonal objects are networks of single polygons (faces) that are connected to form a mesh. This mesh is described to the software as a list of connected points (vertices).

Modifying the position of these points, or vertices, changes the surface contour of the object. Modeling tools in the software allow the user to add, remove and change the link between the vertices to define the polygons that describe an object's surface.

Polygonal surfaces can be used to create virtually any shape and the flexibility of adding more vertices or faces anywhere on the object makes even the most intricate models easy to create. Nonetheless, polygonal surfaces are described with the smallest amount of data, and therefore, can be rendered more quickly, delivering increased speed and interactive performance to the artist. For this reason, polygonal surfaces have a wide range of applications and are the preferred surface type for interactive games and web development.



Figure 13 An example of polygon modeling [14].

2.3.4 Low-poly

A model is considered low-poly when it has relatively few polygons, which also means few faces and vertices. Low-poly models use less computing power, so they update more quickly in viewports and render faster than high-poly models. For this reason, low-poly models are used in real-time games. They are also used even when the final result will be high-resolution output, such as broadcast (TV) or film. In this case, the low-poly model is used during the animation phase, and then a high-poly version is substituted just before rendering. Even with high-poly models, the fewer polygons designer can use without sacrificing quality the easier the model is to work with.

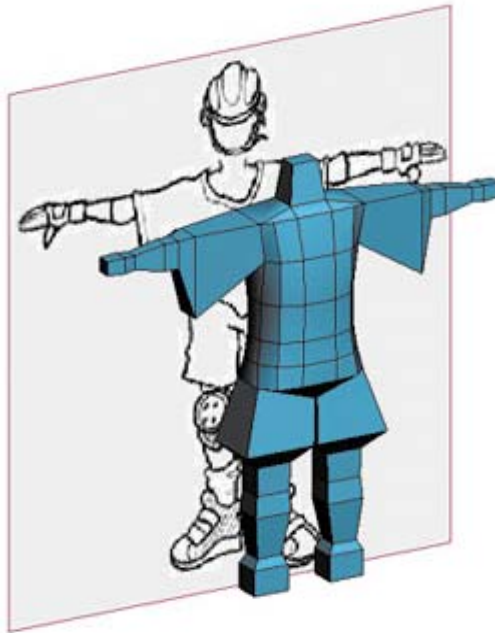


Figure 14 An example of low-poly model [12].

2.4 Texture Mapping

Texturing an object is understood as the process of adding skin and color to an object. A method of attaching color information to a 3D model is by applying a 2D texture image to the model's surface through a process called texture mapping. The 2D texture image applied is just common digital images which use the model's 3D coordinates to specify which parts of the digital image map to which parts of the 3D model's surface. Textures allow 3D models to look significantly more detailed and realistic than they would otherwise.

2.5 Character Rigging

After modeling, designer could animate the character by animating its vertices, but animation goes much more quickly and easily if the character is given a set of controls, recognized as a rig.

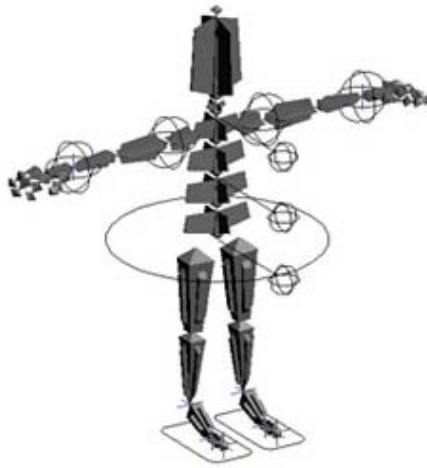


Figure 15 An example of character rigging [12].

2.5.1 Bones

A bone is a special type of object in 3ds max. When designer create a series of bones, they are automatically linked together. In addition, you can automatically add animation controls to the chain as created. Designer will use bones to form the character's skeleton and provide the basis for the character rig.

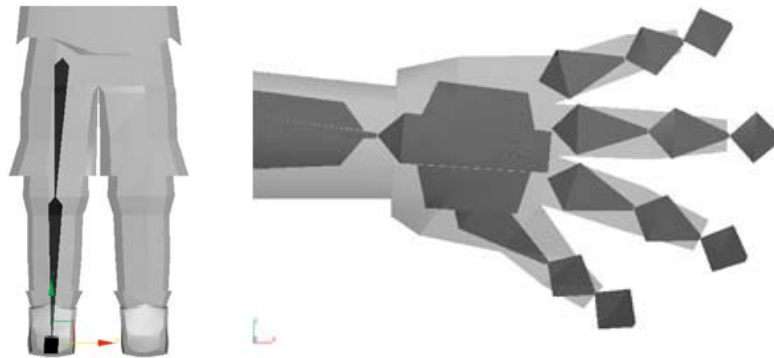


Figure 16 An example of bones [12].

2.5.2 Rigging

A rig is constructed from bones and other objects and parameters. After the character's skeleton is built, designer will add the controls that make it easier to move the character.

2.5.3 Skinning

The skeleton is already in place and rigged for easy movement, but until it is attached to the mesh, it cannot move the character. The process of associating the mesh with the bones is known as skinning. Technically speaking, skinning is part of the rigging process. However, it requires a set of tools and techniques quite different from those setting up controls for bones.

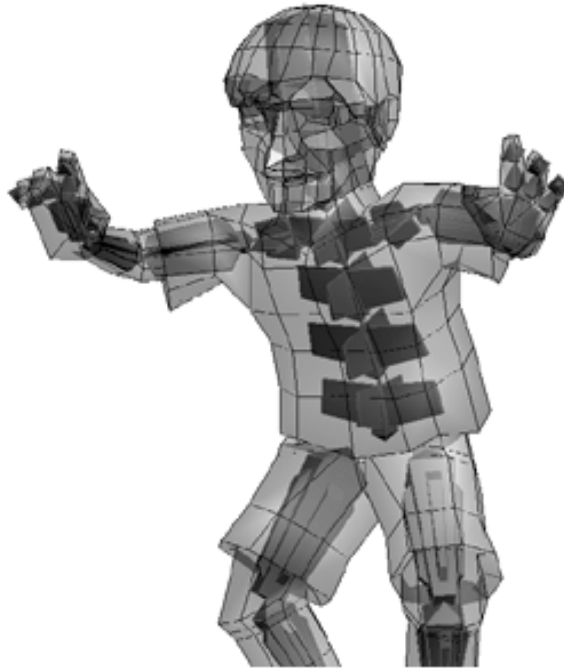


Figure 17 An example of a character completely skinned [12].

2.6 Animation

Animation uses a sequence of still images played back in a rapid chain to create the illusion of movement. Traditionally, animation has always been filmed but computers have made the duty quicker and more economic. The process of creating animation has become more flexible with the computer software allowing artists to preview and test their work before committing to a final sequence and by performing the ‘inbetweening’, so the animator need not animate every frame.

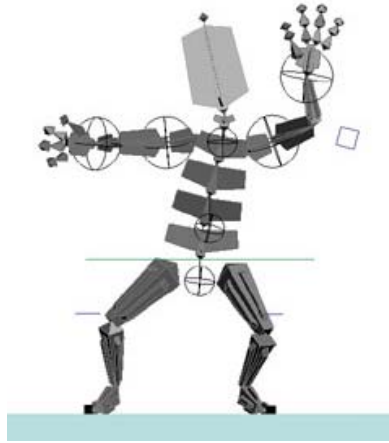


Figure 18 An example of character animation [12].

There are some important terms in creating an animation using computer graphic software such as 3dx max, they are;

- Frame

Frame is generally means a still image in a film or animation.

- Frame rate

The number of frames per second a film or animation is intended to be seen at. NTSC runs at 30 frames per second and PAL runs at 25 frames per second. The higher the frame rate, the smoother the motion will be because there are more frames to be displayed in a second.

- Keys or Keyframe

An animation keyframe or key animator is a single pose or still image within a sequence of animation which occur at an important point or frame in that

sequence. It indicates the timing and number of in-between drawings required.

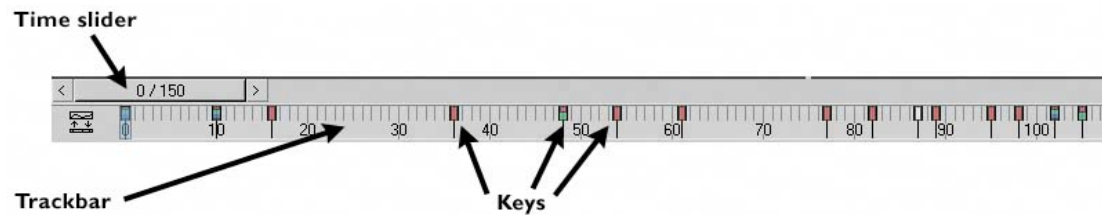


Figure 19 The track view windows in 3ds Max [12].

2.7 Rendering

Rendering is the last step in creating the Computer Graphic work, but it is the step where the scene has to be built such as camera movement. It records the virtual 3D scene into a flat, 2D image, much as a camera records a photograph. By looking through a virtual camera in software to take in surface and light information at a given frame, rendering can make the scene look indistinguishable from reality or as flat as a cartoon. To get to the point where the computer takes over, the camera and render settings have to be set so that we can get what we need from the scene.

2.8 Computer Graphic

According to Edward Angel, Computer graphics is concern with all aspects of producing pictures or images using a computer [15].

The development of computer graphics has been driven both by the needs of the user community and by advances in hardware and software. The applications of computer graphics are many and varied; we can, however, divide them into four major areas: display of information, design, simulation and animation, and user interfaces.

2.9 Computer Graphics Program

3D computer graphic program is a 3D modeler used to visually produce polygonal 3D models. Many 3D modelers can be used to produce models of various real world entities, such as foliage, vehicle, people, etc. They allow designers to build and also modify models through their 3D mesh. Designers can attach, deduct, stretch, and adjust the mesh as they wish. Furthermore, models can be rotated, zoomed in and out, and also viewed from various angles anytime. Most of 3D computer graphics programs also can export their models to files which can be imported into other applications.

The following are some of the most popular and heavily used general purpose 3D computer graphics programs:

- 3D Studio Max

3D Studio Max or 3ds Max is a full featured and most widely used 3D graphics application and animation programs developed by Autodesk Media and Entertainment. It is mostly used by video game developers, movie effects, TV commercial studios, and others.

- Maya

Maya is a high-end 3D computer graphics and 3D modeling software package originally developed by Alias System Corporation, but now owned by Autodesk also. Maya was originally released for the IRIX operating system, and afterward ported to the Microsoft Windows, Linux, and Mac OS X operating systems. An important feature of Maya which is appealing to large studios which tend to write custom code for their productions is the provided software development kit.

- LightWave 3D

LightWave 3D or in short LightWave, is a computer graphics program for 3D modeling, animating and rendering. It was introduced by NewTek as a component within the Video Toaster which is a comprehensive video production tool in 1991 then quickly developed an independent following among a growing number of new animators.

2.10 Waterfall Software Development Life Cycle Model

Waterfall model is a popular version of Software Development Life Cycle (SDLC) model for software engineering. It is often considered the traditional approach to the SDLC which adopt linear and sequential sort of development. Therefore the project is divided into several stages (phases) where the workflow is engaged in sequence, once and only once. Waterfall development has distinct goals for each phase of development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back. The final product is not delivered until all phases are finished.

Waterfall method has many advantages, since it enables shortest development time which theoretically the software can be delivered on time with less cost possible. However, the disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the last stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

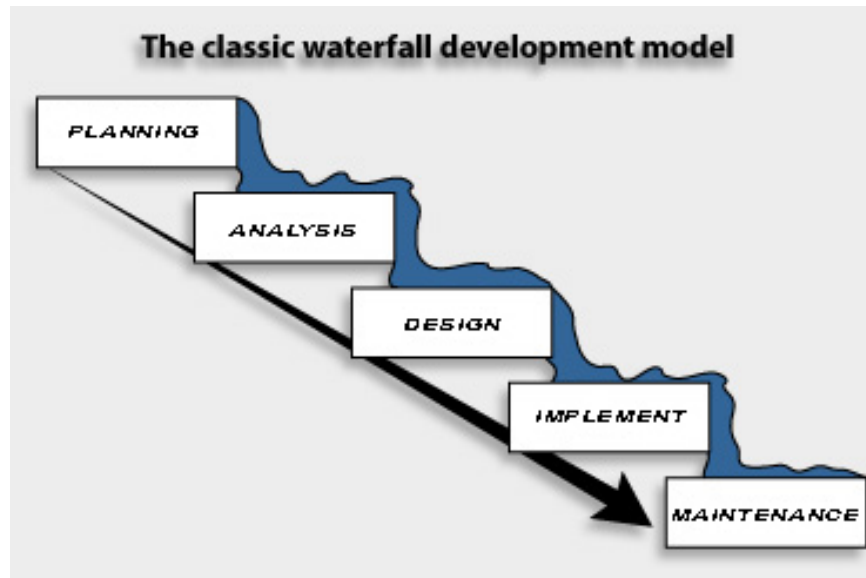


Figure 20 The classic waterfall development model [16].

The model consists of six distinct phases, namely:

1. Planning

In this stage, project team identifies goals to achieve and defines the scope of the project to determine how big the project is. For example, the scope would be defined in terms of the data required in a system developed in the project, the processes to be supported by the system and the system interface with the users. Having determined the scope, the members of the team can estimate the resources for system development and prepare activities required for project accomplishment.

2. Analysis

This phase deals with identifying the problem and requirement of the project. Problem analysis provides for a study and analysis of the existing system, and gives the project team with a more thorough understanding of the problems that triggers the project. The analysis can be conducted through collecting factual information about existing related technologies, and system users concerning the business and perceived problems, causes, and effects.

While requirement analysis provides detailed specification required to support the systems. From the system users, the team collects and discusses requirements and priorities. This information is collected through interviews, questionnaires, and facilitated meetings.

3. Design

The purpose of this stage is to transform the information that is developed during analysis into a design specification for construction that conforms to the target technology that will be used for implementation. The design phase addresses how the technology will be used in the new system.

4. Implement

The purpose of this stage is to construct the system according to the plans and designs from the preceding stages. Implementation corresponds to technology definition and system of the system architecture. It involves a series of activities such as creating program code, testing the system, developing documentation, and utilizing database.

5. Maintenance

Once the system is operating, it delivers the business solution to the user community. It will still require ongoing system support for the remainder of its life and maintenance required to fix any errors, omissions, or new requirements that may arise. The system support consist of some ongoing activities like assisting users, fixing software defects, recovering the system, and adapting the system to new requirements.

