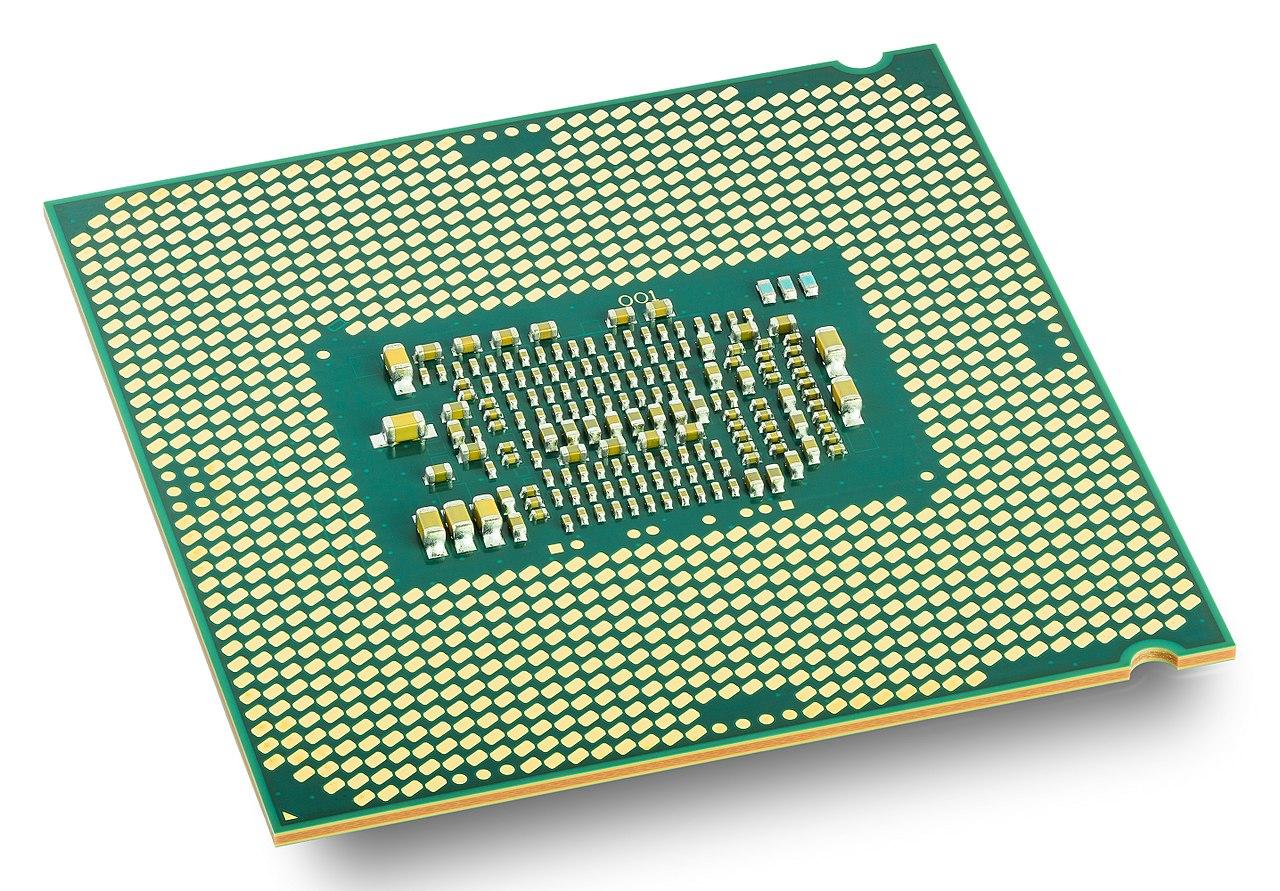
How Software Is Produced Professionally

The modern global infrastructure relies heavily on tasks completed by **software,** which are human-written instructions translated by computers to complete processes based on human needs. You can utilize software right now by turning on your phone and checking the time. The software that supports basic functions such as this on phones and computers is called the **operating system**. Using operating systems, we can download from the internet and operate other software that help start small businesses, research information, and create other pieces of software. This potential for creativity and innovation has driven the evolution of software development for 70 years.

The Early Days

On June 21, 1948, Tom Kilburn of the University of Manchester in England successfully ran the first piece of software ever on the [Manchester Small-Scale Experimental Machine](http://www.cs.man.ac.uk/CCS/res/res20.htm#e) (SSEM)- using punch cards to provide an input instruction for the computer to follow and produce an output. This inspired a bond between writing input-output instructions and developing technology to accept more instructions simultaneously. With the popularization of the personal computer in the 1970s by companies like Apple, IBM, and Microsoft; anyone who could afford it might buy a personal home software processor. (Yost, 2021)

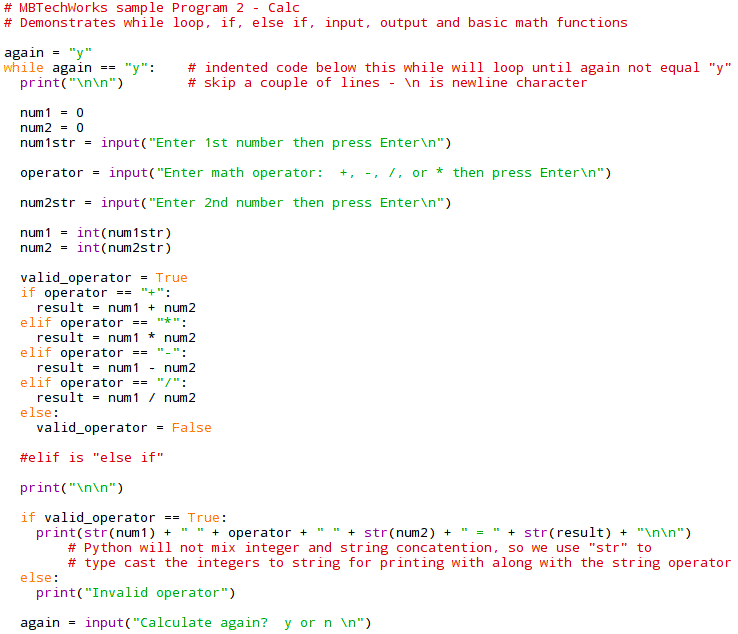
How Do We Use Software?

Now in the 21th century, computer technology has shrunk so much that a room-sized computer from the 1960s can fit in our pockets. The internal components that determine this size and that control the processes within computing devices (i.e. phone, computer, calculator) are called **hardware.** Only a few essential pieces of hardware are needed for software to run. **Microprocessors** and **central processing units** (**CPU**) enable the interpretation of information through electrical currents, just like a brain, and are built with more routes for currents to flow, or *transistors,* when installed in heavily software-reliant devices.

(Wikipedia contributors, 2021)

The **CPU** is the most essential piece of computer hardware made of tiny *transistors,* which act as on/off switches to allow currents of instructions (input) and information (output) to be sent throughout a device.

For any input or output to be stored on a device, information is sent to the **Random Access Memory** **(RAM)**. RAM is another essential piece of computer hardware that often determines the speed at which a device will run since it controls how and how much information can be stored. Software sends and receives information to and from the RAM which then allows the CPU to manipulate information into useful output. (GCFGlobal, 2021)



Writing Code

Suppose we’re writing code for a music application software. First, we choose the language of **source code** to write in, as seen on the right. Source code allows computers to interpret human logic into instructions that provide an output. Examples of source code are Java, C++, Ruby, and Python; as seen on the right, which all have advantages and disadvantages when writing code for a specific usage, such as for a webpage, video game, or calculator software. To create our application, we choose to write in Java since its structure is suited for the application development process. Once we’ve (MBTechWorks, 2018) written our code, a microprocessor or CPU compiles

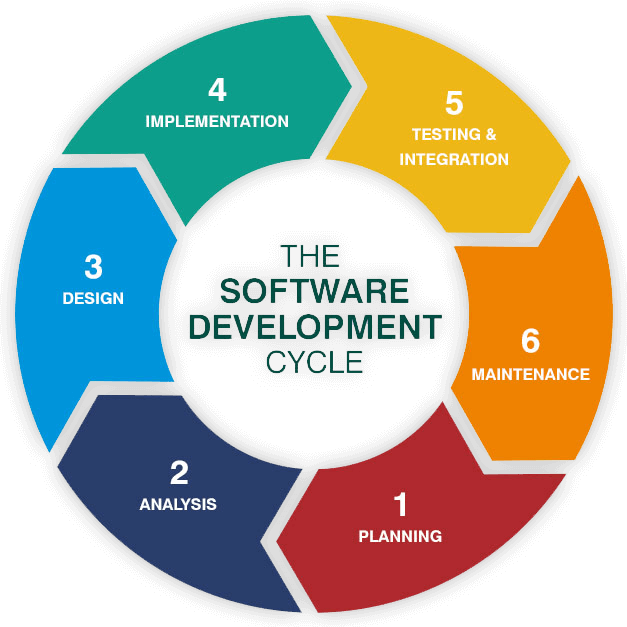
source code into **binary,** which is the only computer language truly understood by computing devices formed by strings of 0s and 1s (i.e. 100 in binary is represented as “1100100”). Conversion to binary is the turning-point where our device recognizes the specific actions we’d like our software to perform.

There are two types of software, application and service, which are defined by how they are used. An **application software** is downloaded by users for a specific purpose, like our music application. A **service software** is the operating system that allows us to download and distribute our application. The beauty of software is that an operating system allows our device to run our music application, which could then run other software possibly written in different source codes.

Why Do We Love Software So Much?

Since the global infrastructure is highly dependent on endless amounts of software running correctly, some of the world’s largest companies are dedicated to the planning, creation, testing, and implementation of software. Software can be written by anyone with a computer and the knowledge of how to, but creating and distributing a piece of software that runs efficiently, on a large scale, and for a long period of time takes teamwork from skilled coders, mechanical engineers, and concept planners. (St. Augustine’s College, 2012)

Microsoft, Oracle, and SAP reigned as the top grossing software development companies in 2020- all netting over $30 million annually and employing over 100,000 employees respectively. (Reiff, 2020) These companies generally are divided into departments with specific tasks to complete during a **software development cycle (SDC),** as seen in the figure below. The SDC provides a path for businesses towards production of high quality software suitable for their infrastructure and consumer needs. (Feoktistov, 2021)

The Software Development Cycle

Let’s recall our music application software example and suppose we followed its development through the various departments of an organization. What steps would be taken and what types of professionals would be needed?

Objectives, required departments, timeframes, and deadlines must first be *determined* by heads of teams and executives to arrive at a logical concept of a music streaming software. An *analysis* of computer hardware requirements and risks is completed by departments needed in the SDC. Once we know our limitations, software architects and engineers *design* the structure of a feasible prototype.

(Feoktistov, 2021)

The design structure will then be *implemented* into source code by software engineers. Once a prototype application is presented, it will evolve through an alpha, beta, and final release phase where the software is updated and fixed to meet financial and architectural requirements. The Quality Assurance team takes over once a final product is approved and *tests* the functionality and unforeseen outcomes to ensure a bug-free product.

Testing should provide a working application that can be distributed to consumers to install and use. With full *integration* reached, we can receive real-world feedback from users that gives the organization vital information for the future of the application. Through this feedback, we carry out *maintenance* which will receive the largest portion of our attention during this SDC. If the organization provides professional maintenance and produces forward-thinking updates for users to enjoy, our application could begin competing with companies, such as, Spotify and Soundcloud, to become a household name in the music industry. (Feoktistov, 2021)

Where Are We Headed?

In 70 years, the evolution of software development has transitioned from using punch cards to Artificial Intelligence capable of writing and giving instructions to other pieces of software. The “new frontier” of software development is focused on the evolution of **Artificial Intelligence (AI)** technology; which is autonomous software written with extensive, predetermined instructions tasked with receiving, interpreting, and outputting information as a human would. AI allows us to drive a taxi-cab or run a factory production line (Simon, 2020) without the need of traditional human laborers. 

(Marsh, 2019)

With robotic technology removing the need for humans in certain job sectors, the global infrastructure must be prepared to deal with the implications of less traditional jobs and more dependence on software running our systems independently. AI allows humans to save time, be more resourceful, and produce more of a desired output which could result in global economic gains primarily for those most impoverished. Businesses and governments will have the capability to reap many rewards that AI and economy experts warn could further increase the current enormous wage disparity experienced globally. With this in mind; executives, software developers, and mechanical engineers have a growing reliance on each other’s skill sets in the drive towards intelligent, efficient, resourceful, and extraordinary technology. (Cope, 2020)

Bibliography

GCFGlobal. (2021). *Computer Basics: Inside a Computer*. GCFGlobal.Org. https://edu.gcfglobal.org/en/computerbasics/inside-a-computer/1/

Cope. (2020, January 15). *What Is the Future of Software Development?* Perforce Software. https://www.perforce.com/blog/the-future-of-software-development

Feoktistov, I. (2021, April 23). *7 Steps of Effective Software Product Development Life Cycle*. Relevant Software. https://relevant.software/blog/7-steps-for-effective-software-product-development/

Marsh, H. (2019). *How Artificial Intelligence Affects Software Development*. Learnhub.Com. https://learn.g2.com/ai-in-software-development

Reiff. (2020). *10 Biggest Software Companies*. Investopedia. https://www.investopedia.com/articles/personal-finance/121714/worlds-top-10-software-companies.asp

St. Augustine’s College. (2012, July 23). *The impact of software*. Software Design and Development. https://sddhsc.wordpress.com/hsc/9-1-development-and-impact-of-software-solutions/9-1-1-social-and-ethical-issues/the-impact-of-software/

Yost, M. (2021, February 19). *A Brief History of Software Development - Micah Yost*. Medium. https://medium.com/@micahyost/a-brief-history-of-software-development-f67a6e6ddae0

Wikipedia contributors. (2021, May 1). *Skylake (microarchitecture)*. Wikipedia. <https://en.wikipedia.org/wiki/Skylake_(microarchitecture)#/media/File:Intel_CPU_Core_i7_6700K_Skylake_perspective.jpg>

MBTechWorks. (2018). *Python Programming Overview*. Https://Www.Mbtechworks.Com/. https://www.mbtechworks.com/programming/python-overview.html

Simon, M. (2020, June 16). *You Can Now Buy Spot the Robot Dog—If You’ve Got $74,500*. Wired. https://www.wired.com/story/you-can-now-buy-spot-the-robot-dog/