My learning about software and large data

Software is a collection of computer data and instructions organized in a particular order. Software is generally divided into system software, application software, and middleware in between. Software does not consist only of computer programs that can be run on computers (by which I mean computers in a broad sense). The documents associated with these computer programs are generally considered part of software. Simply put, software is a collection of programs and documents. It also refers to the management system, ideology, ideological and political consciousness, laws and regulations in the social structure.[1]

Addition, System software provides the most basic functions for computer use, which can be divided into operating system and support software. Operating system is the most basic software.

System software is responsible for managing the various independent pieces of hardware in a computer system so that they can work together. System software allows computer users and other software to treat the computer as a whole, regardless of how each underlying piece of hardware works.

1. The operating system is a program that manages the hardware and software resources of the computer. It is also the kernel and cornerstone of the computer system. Operating systems are responsible for such basic tasks as managing and configuring memory, prioritizing the supply and demand of system resources, controlling input and output devices, operating networks, and managing file systems. The operating system also provides an operating interface that allows users to interact with the system.
2. Support software is the software that supports the development and maintenance of various software, also known as software Development environment (SDE). It mainly includes environmental databases, various interface software, and toolsets. Famous software development environments include IBM's Web Sphere, Microsoft's and so on.

Includes a set of basic tools (such as compilers, database management, storage formatting, file system management, user authentication, driver management, network connectivity, and so on).

Application software

System software is not specific to a specific application domain, whereas application software is the opposite. Different applications provide different functions based on the user and the domain they serve.

Application software is software developed for a specific purpose. It can be a specific program, such as an image browser. It can also be a collection of closely related programs that work together, such as Microsoft Office. It can also be a large software system consisting of many independent programs, such as a database management system.

Nowadays, smart phones have gained great popularity. The application software running on mobile phones is referred to as mobile phone software. The so-called mobile software is the software that can be installed on the mobile phone to improve the shortcomings and personalized of the original system. With the development of science and technology, the functions of mobile phones are more and more powerful. It's not as simple and rigid as it used to be, and it's developed to be comparable to palmtops. Mobile phone software is the same as a computer, when downloading mobile phone software, you also need to consider the system you buy the phone installed to determine the corresponding software. Mainstream mobile Phone systems include Windows Phone, Symbian, iOS, and Android.[2]

For lager data， McKinsey Global Institute defines it as a data set that is so large that it is beyond the scope of traditional database software tools in terms of acquisition, storage, management and analysis. It is characterized by massive data scale, fast data flow, diverse data types and low value density.[3]

About why it is important,here are the reasons.

Take advantage of social influence

For many traditional industries, transformation is imperative, so how to transform? Many of them are transforming through O2O mode, embracing the Internet, attracting customers online and letting them consume offline. Then we can make full use of social influence in offline publicity.

2. Good user guidance service

In many software, documents, data, we can see there are instructions that it will be very helpful in improving the software user's experience, a lot of people think the documentation as professional as possible, actually otherwise, you can try in these documentation and some introductory and persuasive words, it can help customers find what you need better.

Rare things are precious

New things are always popular with people, regardless of the star, TV dramas, and even new inventions, and so on, there may be some difference for the software, software is developed for customer service, and customers through search engines to find you, so in this a piece of software development, we not only need to do is novel, also make the search engine is more like you.

4. Excellent design and service

As for software, beautiful websites are always favored by people. If one software uses graphic form and the other software uses text form, which one do you prefer to read? As for the service, an enterprise should have a perfect service system, only in this way can the user churn rate be reduced to the maximum extent. High-quality service is also a way to reduce the jump rate.

5. Consider what the user wants to achieve

Why do people use your software? What services do users need when they come here? These are the problems you need to consider before developing software, if a user into your software, but instead lost, can not find what they need, believe that any enterprise does not need to develop their own software such a situation.

6. Use nudges to influence user behavior

Actually WeChat did very well on this point, from the public to the small program, from small program to since the rise of media platforms, this is a change in the user's online behavior, whether it is or the public, or the media, the automatic push some news for you, every day in imperceptible influence the user's behavior on the Internet.[4]

1. E-commerce: We believe that the application of big data in e-commerce has been common. E-commerce platforms such as Taobao and Jingdong make use of big data technology to analyze user information, so as to push products that users are interested in and stimulate consumption.

2. Government: "smart city" has been tried to operate in many places. Through big data, government departments can perceive the needs of social development and change, so as to provide citizens with appropriate public services and resource allocation in a more scientific, accurate and rational way.

3. Medical field: The medical industry assists Y1 in clinical decision-making, standardizes diagnosis and treatment paths, and improves work efficiency through clinical data comparison, real-time statistical analysis, remote patient data analysis, and medical behavior analysis.

4. media field: media related enterprises through the collection of all kinds of information, classification, screening, cleaning, in-depth processing, to achieve accurate positioning and grasp of readers and audience Ge Xinhua needs, and tracking users' browsing habits, continuous information optimization.

5, security field: security industry can realize video image fuzzy query, fast retrieval, precise positioning, and can further mine the value information behind massive video surveillance data, feedback connotation knowledge to assist decision-making.

6. Financial sector: Based on user portraits, banks can accurately locate user groups according to their age, asset scale, financial preferences, etc., and analyze potential financial service demands.

7. Telecom: The telecom industry has a huge amount of data. Big data technology can be applied to network management, customer relationship management, enterprise operation management, etc., and the data can be commercialized externally to achieve independent profits.

8. Education: Through big data learning analysis, we can create a tailor-made personalized course for each student, and provide a challenging but not gradually boring learning plan for students' years of study.

9. Transportation: Big data technology can predict the future traffic conditions, provide optimization plans for improving traffic conditions, help the transportation department to improve the control ability of road traffic, prevent and alleviate traffic congestion, and provide more user-friendly services.[5]

large data technology can not only improve the efficiency of data utilization, but also realize the reuse and reuse of data, thus greatly reducing transaction costs and enhancing the space for people to develop their potential. people can low-cost or zero cost of things information holographic vertical historical comparison and horizontal reality comparison. Big data technology itself can not only rapidly evolve into an emerging information industry, but also link with cloud computing, the Internet of things and smart engineering technologies to support a new era of information technology.

I am interest in the phone's app which belong to software.

Everything starts with an idea. This might be your idea or an idea that a client has approached you with. Ideas are great, but they're also a dime a dozen. The sooner you realize that ideas are nothing but passing phantoms of something that might one day turn into a product, the better you'll be able to handle this phase.

We tend to put way too much stock in ideas, as getting the idea 'right' is far less important than people think. Ideas are sealed up and protected by NDA's, paraded around in pitch decks, and tend to take on a very defined state much too early.

Keeping your idea malleable and changing for as long as possible is, in my experience, much healthier and leads to far better end-results. Ideas should be subjected to a healthy dose of Darwinism before being pursued – survival of the fittest. You want your idea to evolve the best version of itself, and to that end, it can make sense to talk about a circular process within this phase.

Idea workshops are great ways to force the evolution of your ideas. You can use things like Trello to track aspects of your idea in an environment where you can move around and prioritize concepts. Collaboration helps promote the strong aspects of the concept, the ones that resonate with participants. At the same time, collaboration helps identify and eliminate what is detracting from the idea.

Next up, or as is sometimes the case, alongside, is the development of the app. In an ideal world, the person responsible for developing the app has been part of all previous phases, chiming in with his or her experience, deliberating on the difficulty level of implementation of various proposed designs, and discussing best practices in terms of structure, tools, libraries, and so on.

I am familiar with the desire to clearly separate development from design, both from an organizational and a cultural perspective. It is my clear conviction that the best products are built by teams made of multiple professionals from various disciplines who have a mutual understand of each other. Development shouldn’t be devoid of a design presence and design shouldn’t be without development know-how.

There’s an obvious balance to this. Designers probably shouldn’t have much to say in the choice of an API implementation, like developers probably shouldn’t get to veto a color scheme. However, during a wireframing session a developer could save a designer from making a disastrous proposal that would cause time for implementation to increase tenfold. Likewise, a designer overlooking implementation of navigation could steer the interaction towards a much more enjoyable experience that better fit the consistency and feel of the app. Use your best judgment but, don’t rob your team of their combined expertise by putting up imagined barriers.

A lot could be written about the iterative nature of development as well, but once again, I will let someone else to write that article.

Iterate #

The real truth that seems to catch many people off guard is that you’re never actually done designing. In most good projects, designers have product ownership from spec to ship. You don’t want design becoming a relay race where you hand off something to another department or group of people where you don’t have a say. Even just listing the individual steps like I’ve done, I run the risk of misleading you, as it can very easily be understood as a progression that runs from A to B. Designing apps, or anything for that matter, is rarely a straight line or a clear succession of stages.

While our tools, as well as our products, have changed a lot over these past years, the underlying process of making apps remains largely the same.

1. Get an idea
2. Write it down
3. Build a prototype
4. Enter into the dance between design and development until something comes out of it

As you progress down this narrowing funnel of bringing an app through development, you make assumptions. Then, you challenge and revise until some nugget of truth makes it into the next stage (or the next build).

People tend to think about building apps the way they think about building a house. First, you lay the foundation, then the walls come up and the appliances are installed. It seems straightforward. There is a blueprint, a design, and a team building it. This fallacy is the source of much grief in the world of making software. It’s why clients expect you to be able to tell them how much their idea costs to develop. It’s why estimates are almost always wrong, and frankly, why we have so many terrible products. It implies that we know the outcome of the process – that right from the start we're working in a mostly controlled environment with a clearly defined goal.

But, if the process and the stages outlined above teach us anything, it is that we don't and we shouldn't have one. The process is there to help us explore the potential by challenging our assumptions and iteratively execute each step – to bring the best nuggets from the idea into the hands of people.

Rather than building a house, designing apps is probably more like composing a symphony. Each profession a separate instrument. In the beginning, it sounds odd, hollow and out of tune. Slowly, however, as we move through the acts and apply our experience and skill, iteratively it finds its direction and becomes some version of the music described in the original idea.[6]

Finally,let's talk about the open source about app.

Open source has a lot of advantages. It's a flexible standard, it helps the community, and the community can help open source projects grow. Android is one of the most popular and more powerful open source projectsof all time. Those who like seeing the code that runs on their phones are in luck. There are plenty of excellent open source apps on Android in addition to the OS itself.

Here are some of the best open source apps for Android.

1.Firefox
2.FreeOTP
3.Lawnchair Launcher
4.Nextcloud
5.Open Camera
6.OsmAnd
7.Phonograph
8.QKSMS
9.Simple Mobile Tools
10.VLC[7]

Firefox is one of the most popular open source projects out there. The popular browser has mobile apps as well as desktop apps, obviously. It covers all of the basics, including cross-platform syncing, privacy browsing modes, add-ons, bookmarks, and that sort of stuff. Firefox underwent a major change in 2018 with better and faster browsing. Google Chrome is technically mostly open source, but Firefox is the real deal for open source browsers.

FreeOTP is a two-factor authentication app. It works like Google Authenticator or Microsoft Authenticator. You set it up and it provides security codes for login. It supports both TOTP and HOTP protocols and should work with all websites with support for those protocols. The app is free, open source, and maintained by Red Hat. Yes, we're talking about the Linux distro Red Hat. It hasn’t had an update in a couple of years, but the source code shows activity as of a few months ago so it may get an update sooner or later. There is also andOTP, another decent open source authenticator app.

Lawnchair Launcher 2 is one of the newer open source apps on the list. It’s a launcher with a stock Android theme. It closely resembles the Pixel Launcher, except with more features. They include Google Now integration (with an add-on), icon pack support, variable icon size, a blur mode, and other customizations. This one is still in beta. However, we feel comfortable recommending it to almost anybody. It’s also free, open source, and highly functional. It’s great for those who like a minimal launcher experience, but still want some customization features. The developer is working on the beta for the second iteration of this launcher, so do beware of bugs for the time being.

In a word,software and large data are very important to people’ life. We need to keep developing them .That’s all, thank you !

Work cites

[1]Overview of software life cycle model and application criteria. Sina Column [Citation date: 2015-01-21]

[2]Easy Language 4.0 Beta 6 release. CnBeta software news. 2005-06-07[Citation: 2014-09-29]

[3]Big data era requires big data thinking. Big Data in China [Citation Date: 2015-11-03]

[4]www.sohu.com, [accessed Sep 17, 2021]

[5]http://www.qianjia.com/zhike/html/2019-12/18_17785.html[accessed Sep 17, 2021]

[6]Michael Flarup,What You Should Know About The App Design Process , www.google.com, [accessed Sep 17, 2021]

[7]Joe Hindy,10 best open source apps for Android,www.google.com, [accessed Sep 17, 2021]