Sources of Software

The origins of software

- Many software is not written at once or by our own. Sometimes we write small separate applications and combine them together. Sometimes we use 3rd party software.
- Companies specialize in several areas. That is why using 3rd party software instead of developing everything in-house is sometimes more beneficial for the company.

Sources of software

- **Information technology services firms.**
- **Packaged software providers.**
  - Instead of providing their applications online, they package their software. Microsoft Office is an example to it. Although in the recent years they provide online services, we used to buy them on CD with licensing.
- Vendors of enterprise-wide solution software
- **Cloud computing**
- **Open source software**
- **In-house software**
  - These software are built by the companies themselves.

First software

- General Electric was the first company to build an administrative information system. It was for **payroll**.
- Before that, people were building software in-house. Everyone was building different software to solve a **single** problem. Of course, at those times software were primitive and small in scale.

Outsourcing

- It is a general term and not only used for software.
- When an organization develops or runs a computer application **for** another organization, it is called *outsourcing*.
- Shell company outsourced all its SAP-based HR and Payroll Management to a company called *Accenture*.

Why outsource?

- You expect them to be professionals. You don't deal with educating them.
- You don't deal with hiring people specifically for that job.
- They are working for multiple companies and therefore provide better pricing.
- Frees up internal resources. You can use your developers for something else.
- Reduce the time to market. Development would be faster.
- Political reasons.
    - Firing is much easier!

Example

- Car rental companies can be seen as a good example. If you buy a car, there are a lot of stuff to take care of such as service, insurance, etc
- Car rental companies take care of all these and they even replace your car every year or two.
- You should always check if outsourcing is beneficial for the company. It is possible that someone actually solved the problem you have.

Information Technology Services Firms

- Many companies needs software, but they are not a software company.
- It is also possible that they don't have an off-the-shelf solution available.
- Well, they need a newly built software!
  - However, they have couple of options.
- They may choose to hire people to build that software, but is it really necessary?
  - Here, ITS firms help these companies.
  - These firms helps to build software or supply a pre-built software for the needs of the company.
- Microsoft, Oracle, Amazon are such companies.
  - Office programs help us a lot.
  - SAP is a good example, etc.

Packaged software providers

- Microsoft.
  - Almost 90 percent of its income comes from packaged software such as Windows and Office.
- Oracle
  - Creates enterprise systems but the main income is from database systems.
- SAP
- These products are not personalized. It is said that it can only meet the requirements of 70 percent at most.
- In *turnkey* systems, if the number of customers demanding a change is large, they can improve the software or add functionalities.

Enterprise solution software

- ERP
  - Consists of series of integrated modules
- In the traditional approach, different systems will be used in different areas. There would be a billing software for accounting, inventory system in the warehouse, etc.
- In ERP, these modules are integrated in a **single** software. The aim is to provide a **seamless** experience.
- SAP is the most popular ERP software.

Cloud computing

- Software runs in another server or computer. We are connected to it **remotely**.
- We don't deal with installing, maintaining or upgrading. We don't deal with hardware issues. The provider deals with them.
- Google Docs, Office 365 are good examples. **Paraşüt** is another example, it is an online accountig firm. **Microsoft Azure** and **AWS** are also well-known examples.

Cloud computing (cont.)

- It frees up internal IT staff.
- Gaining access to applications are faster and lower-cost.
- However, people are concerned about **security**.
  - We are storing **our** data on computers that we don't have **physical** access to. Other people may have unauthorized access to company and customer information if these are not encrypted.
    - Usually they are not.

Cloud computing (cont.)

- Reliability is another reason.
- In cloud computing, there is a *complexity* because we are talking about a network of networks. Virtualizaton is heavily used in cloud computing.
- Nonetheless, we should always take frequent backups.

Open source software

- Freely available but there are different licenses.
- Keep this in mind.
  - If we are downloading a pre-compiled program, it doesn't matter whether it is open source or not. We cannot know if the program is compiled with **that** code.
  - We should be able to compile that program ourselves. Otherwise, we cannot be sure.

Open source software (cont.)

- They generally make money by maintenance or donations.
- They can also create different versions of the program.
    - A free one with limited functionality and a paid one with more functionality.
- Many programming languages are open source. For example you can see how Python is coded and the source code for that easily in Github. **Linux** is also open source and the code can be read.
- If you are using open source software, you should be careful and watch the maintenance because of **software dependency**. Maybe the program you rely on is not maintained anymore? It is a risk in open source.

In-house development

- Usually, it is not that feasible. However, sometimes you must do them.
- If you are startup, you may choose to use 3rd party software to decrease the time you get to the market.
    - However, as you improve and grow, it is possible that this software can't solve your problems. In that case, it is feasible to create the software in-house.

Deciding which to choose

- **Viability** of the vendor is very important. Are you sure they are going to keep on maintaining the software? What happens if they close the company?
- **Cost** is very important. Compare in-house with 3rd party.
- **Function**: How well the software is adapted to our work? Can it do everything possible or how well can it do it?
- **Documentation**: How is the documentation of the software? Not everyone is tech savvy. Especially if the software is complex and there are lots to learn, how can they learn it? How are you going to educate the users?