This sheet is a handout material from Udemy course:
Essentials of Software-as-a-Service (SaaS) Business.

1.2 Defining Software-as-a-Service

Welcome to the second video of the course where we are going to explore the main characteristics of the Software-as-a-Service model by comparing it with the traditional way of software delivery.

Aspect	Traditional Custom-Made Software	**Software-as-a-Service**
Accessibility	● access limited to specific devices or locations	● access from any internet-connected device
Single-tenancy vs. Multi-tenancy	● single-tenant architecture: each customer utilizes a single instance of a software	● multi-tenant architecture: a single instance of the software serves multiple customers
Updates and Maintenance	● user responsible for software updates and maintenance	● updates and maintenance handled by the provider
Customization and Scalability	● highly customizable but less scalable	● configurable features for diverse needs, with shared core functionality
Pricing Structure	● high initial costs for licenses and infrastructure	● lower upfront costs with subscription-based pricing
Data Management and Customer Relationship	● local or dedicated server storage, transactional vendor relationship	● centralized cloud data storage with robust security and support
Deployment Speed	● slower deployment and fixed capacity	● quick setup

Accessibility

- *SaaS: Access from any internet-connected device.*
- *Traditional: Access limited to specific devices or locations.*

Traditional software requires installation on specific hardware like a personal computer of a user. In contrast, SaaS applications are hosted on Cloud servers and delivered through the Internet. Users can access the software from any device with an Internet connection. This really enhances work flexibility and mobility. We can access our work documents on a laptop at a café as easily as from our office.

Single-tenancy vs. Multi-tenancy

- *SaaS: Multi-tenant architecture - a single instance of the software serves multiple customers.*
- *Traditional: Single-tenant architecture - each customer utilizes a single instance of a software.*

In the case of traditional software, each customer utilizes a single instance of the software, which is known as a single-tenant architecture. This means that each customer has their own dedicated resources and customizations. In contrast, most SaaS applications use a multi-tenant architecture, where a single instance of the software serves multiple customers. This allows for better resource utilization and easier scalability, as the provider can manage and update the software for all customers simultaneously.

Updates and Maintenance

- *SaaS: Updates and maintenance handled by the provider.*
- *Traditional: User responsible for software updates and maintenance.*

When using a traditional software, the customer is solely responsible for updates, maintenance, and security patches. This can be both time-consuming and can lead to inconsistencies in software versions or potential security vulnerabilities if not managed properly. SaaS models alleviate this burden. The provider of the SaaS product manages all of these aspects, so there are no actions needed from the customer's side. Moreover, continuous updates and new features can be rolled out more frequently, keeping customers at the cutting edge of technology and adapting quickly to market changes. For example, if you use Google Workspace, you can be sure that you always have the latest features and also protections without manual updates from your side.

Customization and Scalability

- *SaaS: Scalable and configurable features with shared core functionality*
- *Traditional: Highly customizable but less scalable.*

Although custom-made software can be deeply customized to fit specific business needs, this customization often comes at the expense of scalability. SaaS solutions prioritize scalability. Since your business needs can change fast - cloud computing adapts just as quickly. Let's say your e-commerce site suddenly gets more visitors - your cloud service automatically adds more resources to handle the traffic, ensuring a seamless shopping experience. Businesses can easily scale up or down based on current needs without significant redevelopment. Also this model allows for significant customization. Suppose that your company utilizes a project management application hosted in the cloud. There you can configure user permissions based on roles within your team. This means you can decide who can view, edit, or manage certain projects and tasks.

Pricing Structure

- *SaaS: Lower upfront costs with subscription-based pricing.*
- *Traditional: High initial costs for licenses and infrastructure.*

Then we have the pricing structure. For traditional software, this typically involves significant upfront costs for licences, hardware, and implementation, then periodic expenses for upgrades and additional licenses. SaaS typically utilizes subscription billing models that allow for predictable budgeting. These models take into account various features such as pricing amount freemium vs. premium, charging based on number of users or pre-set frequency of billing that can be for example monthly, annually payments. We will cover the widely used models in detail later in this course.

Data and Relationship

- *SaaS: Centralized cloud data storage with robust security and support.*
- *Traditional: Local or dedicated server storage, transactional vendor relationship.*

Next we have two aspects in which traditional custom-made software and SaaS differ. When utilizing a traditional software installed locally on the personal computer, the data is stored locally on a personal computer or on dedicated servers belonging to the user's company. If something happens to your personal computer, your data may be lost. If the data is stored or on a dedicated server belonging to the company that the user is working for, that company needs to properly manage backups and recovery which can be resource-intensive. On the contrary, if you utilize a SaaS product, the data is stored centrally on the cloud providing better data security, backups, and recovery processes.

Regarding the relationship with customers, companies that offer traditional software often have a transactional relationship with customers, focusing primarily on the initial purchase and periodic updates or support calls. In contrast, the SaaS model prioritizes ongoing support and collaboration, making it an integral part of the customer experience.

Deployment Speed

- *SaaS: Quick setup.*
- *Traditional: Slower deployment and fixed capacity.*

And finally, deploying traditional custom-made software can be a lengthy and complex process. The Software needs to be installed on each device of customers who will be utilizing it and configure it to meet the specific needs of the user or company. SaaS, on the other hand, offers rapid deployment. The software is simply ready to use almost immediately after subscription.