Uday Chhina
A01210638

Guided study Q's

1. **5 comps of IS**: hardware, software, data, people, and processes.
2. **Information system**: can be defined as a set of interrelated components that collect, store, process and distribute information to support decision making.
3. **Process**: "A process is a series of steps undertaken to achieve a desired outcome or goal."
4. **SDLC**: Software Development Life Cycle, **purpose**: to manage large software projects associated with corporate systems running on mainframes.
5. **Stages of SDLC**: 1. Prelim Analysis – what is the problem that needs to be solved. 2. System analysis – system analyst + stakeholder figure out requirements, no programming. 3. Sys Design – Business reqs to technical reqs. 4. Programming – write code. 5. Testing – code is put through various tests. 6. Implementation – implemented in company, users are trained, conversion, etc. 7. Maintenance – the system is maintained.
6. **Agile**: constant evaluation and development by developers and constant evaluation by customers. **Iterative** means repeating a process again and again while analyzing the outcomes to achieve a better product.
7. **Competitive advantage**: the ability of a company to get profits that are above the average in the industry. Two things affect this: cost advantage and differentiation advantage.
8. **What does value chain help a company achieve**:

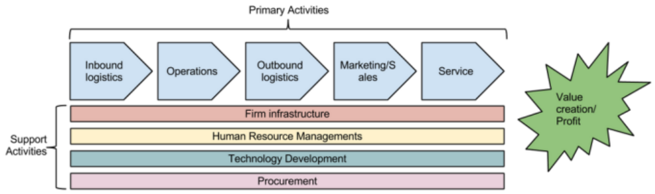
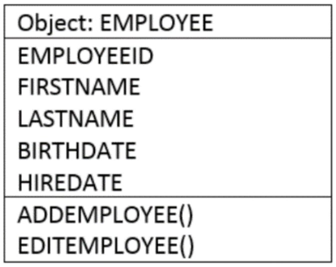


Diagram of Porter's Value Chain (click to enlarge)

value chain helps a company achieve value and therefore profits. Value is created through the value chain – series of processes to produce a product or a service.

9. **Feasibility**: how easily can something be done. Technical, economical, and legal feasibility. The feasibility analysis studies how easy it is to complete the project and if it should be initiated. Basically answers the question if "should we do this project?"
10. **OPEX and CAPEX**: OPEX is the operating cost. CAPEX is the capital expense. The become part of the feasibility analysis of the project. Help us decide is the project should be initiated or not.
11. **Requirements defined in** prelim analysis, system analysis, system design, programming. **Requirements** are things that a business needs for their information systems.
12. "This approach to software development is very structured and risk averse, designed to **manage large projects** that include **multiple programmers and systems** that have a **large impact on the organization."**
13. **Functional Requirements:** The requirements that answer the question of that the system needs to be able to do.
14. **Non-functional requirements**: The requirements that answer the question of how the system needs to be able to meet the functional requirements.
15. **Process**: A series of tasks that are completed in order to achieve a goal. **Business process**: a process that is focused on achieving a goal for the business.
16. **Strategy**: The long-term goal or roadmap of an organisation and how the organisation intends to reach it. **Competitive advantage**: see point 7



Object: EMPLOYEE

EMPLOYEEID
FIRSTNAME
LASTNAME
BIRTHDATE
HIREDATE

ADDEMPLOYEE()
EDITEMPLOYEE()

17. **OOP** is valuable because it allows for the program to respond directly to user input instead of sequentially executing code.

Classes have a name, attributes, and behaviours.

18. **Advantages of the buying software**: 1. Less expensive, 2. Available much more quickly that building it(up and running in a few days), 3. Already been tested and many bugs worked out. **Disadvantages of buying software**: 1. Same software by competitors, not able to distinguish. 2. Less customization.
19. **Blurring the lines between build and buy**: a company might build software from the ground up but then supplement it using functions and software from other companies. Ex. Google Maps API service. This is not as clear cut as buying or building software, they company ends up building some parts and buying others.

Review Questions

Point of BSA: Business systems analysis – to analyse the information systems of a business to the purpose of making better strategies for a higher value product and eventually, better profits.

What are information systems: see point 1 and 2

What **initiates the SDLC**: The preliminary analysis to solve an existing problem. Or a problem that needs to be solved through software.

What is the **output of the SDLC**: The output of the SDLC is a solution of the problem that started the SDLC.

What is a **CBA**: Cost Benefit Analysis: calculating the costs vs the benefits of a product/service or a system.

What is the **top line**: The money earned by selling products and/or services.

What is the **bottom line**: the profit (revenue – expenses)

What is an **investment**? – (The money spent on assets, products, systems, etc for an expectation of a profit or material result) IT, if we spend money on an investment, we call it an asset. We are interested in Information systems. WE do it because it is expected that there will be a ROI.

Organizations aren't always choosing the lowest cost option (expenses). also looking at the return on the investment

What is **ROI**? - the money we are hoping to get back from an investment, like an information system.

What is **TCO**? **total cost of ownership** - acquisition and operation of assets. All the potential costs of having assets - the price to acquire it and operate it. Example - delivery van (acquisition - buying the van, operation - driver, gas, maintenance, insurance, parking when not in use, etc.)

What are **requirements**? - the things that businesses need from their information systems.

What are **functional and non-functional requirements**? Functional is the 'what'. if you can imagine doing it in the LMS, then it is a functional requirement. Non-functional is NOT EQUAL TO "not important". Non-functional: 1. Performance, 2. Reliability, 3. Availability, 4. Security, 5. Maintainability

What is **scalability**? How does it relate to requirements? - scalability is performance, reliability, and availability. ability of our systems to meet the changes in the business flow. scalability is a non-functional requirement.

Swimlanes: 6 parts: 1. The lanes: each actor works within their lanes, 2. Rounded rectangles (or circles): terminators, 3. Arrow: connector – relationship, 4. Rectangle: process, 5. Diamond: decision, 6. Rectangle with parallel lines: sub process.

What is a **business model**? "a company's plan for making a profit. It identifies the products or services the business plans to sell, its identified target market, and any anticipated expenses." – Investopedia

What is the **happy path**? anything that is a business process. like making a big mac how it was defined in the tutorials. if you chose to not get onions, that is not the happy path, but it is still okay since that is a business decision.

What is an **actor**? A role played by a user or any other system that interacts with the subject of the use case/swimlane.

What is **a class diagram** in UML? what are the three parts to a class diagram? A class diagram represents an object. Class diagram is a diagram that shows the different parts of the class. Name, methods (behaviours), and attributes.

What simple **memory aid** can we use to tell if something is a class? - if you can use 'the' in front of it in a meaningful way.

Use case diagram: rectangle: the system, actor: the things that interact with the system – can be human or non human., use case: features or outcomes of value, the lines: the connection between actor and use case.

Make

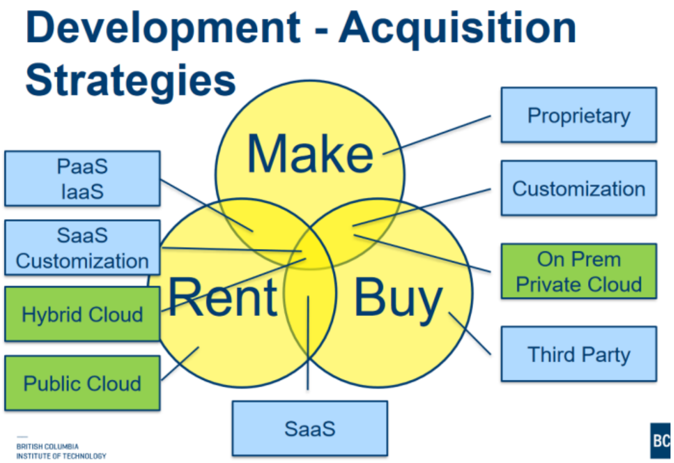
- When an org has custom or proprietary — or strategic - requirements, e.g., something they can't (or won't) buy from a 3rd party
- More expensive to implement and maintain and requires the organization to have systems development and maintenance capabilities "in house"

Buy

- Standard functionality (i.e. every customer gets more-or-less the same), often with built-in integrations (e.g. Microsoft 365 or Salesforce)
- Lower acquisition costs, faster to deploy and can be less expensive to maintain than in-house - but harder, more expensive to upgrade and migrate

Rent

- Cloud, subscription models and outsourcing of IT and dev capability
- Standard functionality across all tenants — often with elastic scalability
- Can be expensive and harder to customize, but has advanced capabilities
- Important - no custody of data



Development - Acquisition Strategies

Generally, why would the **TCO of a cloud option tend to be lower than in house** – The TCO of a cloud option is generally cheaper because the cost of the cloud service is being spread over all the customers that are also using the cloud service. Setting up in house servers and data storage centers would mean 100% responsibility of the cost.

What is the key reason why a company would **consider building their own information system** – When the company wants to differentiate themselves. If the company wishes to give itself competitive advantage, they might design their own IS. The companies also consider the ROI on the implementation of the system rather than just the investment cost. They might also wish to keep their trade secrets private.