PROJECT MANAGEMENT

MODULE 2 – Winter 2023

HUMBER

WE ARE HUMBER

AGENDA

- Project Feasibility (Technical, Operational, Economic, Schedule)
- Identifying Roles & Assigning Responsibilities
- Stakeholders
- SWOT Analysis
- System Architecture
- Development Lifecycles
- Next class

Vision, Mission & Values

Vision

The direction that an organization intends to take in-order to develop and strengthen its business. The intent of a vision statement is for Senior Management to clearly articulate the management's aspirations to stakeholders and focus the energies of all organizational resources in a common direction

"A car in every garage"

- Henry Ford's vision

"A personal computer in every home running Microsoft Software"

- Microsoft's vision

"To be the company that best understands and satisfies the product, service and self-fulfillment needs of women – globally"

- Avon's vision

Where we are going??

Vision, Mission & Values

Mission

Present business and purpose; it focuses on the current business activities and demonstrates the reason for being!

Who we are?
What we do?
Why we are here?

"Organize the world's information and make it universally accessible and useful"

- Google's mission

"To inspire and nurture the human spirit – one person, one cup and one neighborhood at a time"

- Starbuck's mission

"Leader in global beauty, Women's choice for buying, Premier direct-selling company, Most admired company, Best place to work, To have the largest foundation dedicated to women's causes"

- Avon's mission

Vision, Mission & Values

Values
Beliefs, traits, and behavioral norms that company resources are expected in conducting company's business.

What is our creed?

- **Excellent service to client s and each other**
- **Working together to succeed**
- **Personal responsibility for high performance**
- **Diversity for growth and innovation**
- **Trust through integrity in everything we do**

- RBC's values

- **Initiative**
- **Equality**
- **Trust**

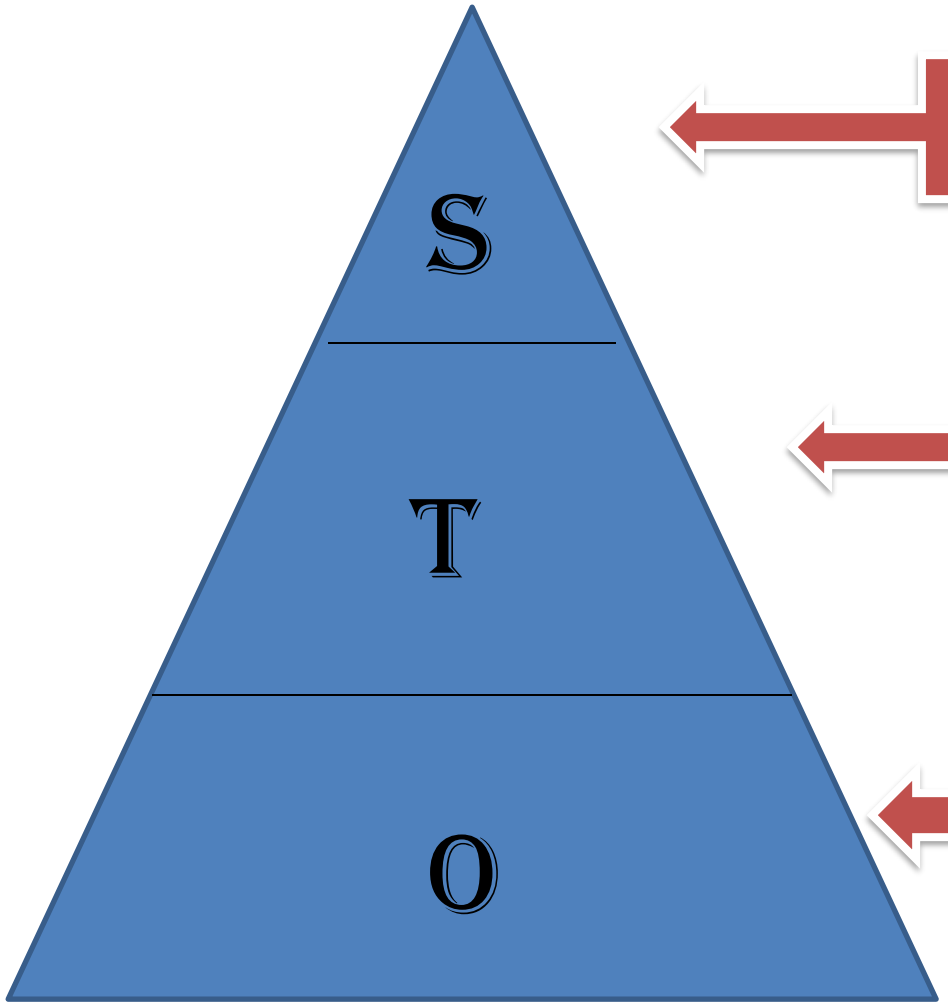
- Honda's beliefs

- **Trust**
- **Respect**
- **Belief**
- **Humility**
- **Integrity**

- Avon's values

WE ARE HUMBL

STO Model (Strategic, Tactical & Operational)



What to do?

How to do?

What day to day process and policies?

STO Model

- Strategy is a series of concrete actions that an organization needs to undertake to strive towards its mission, and ultimately its vision

- Fundamental responsibilities of organizations are twofold:
  1. Create value for the owners (or shareholders)
  2. Carrying out the mission in an ethical manner

- Projects that are being carried out by the organization must contribute to the above two factors.

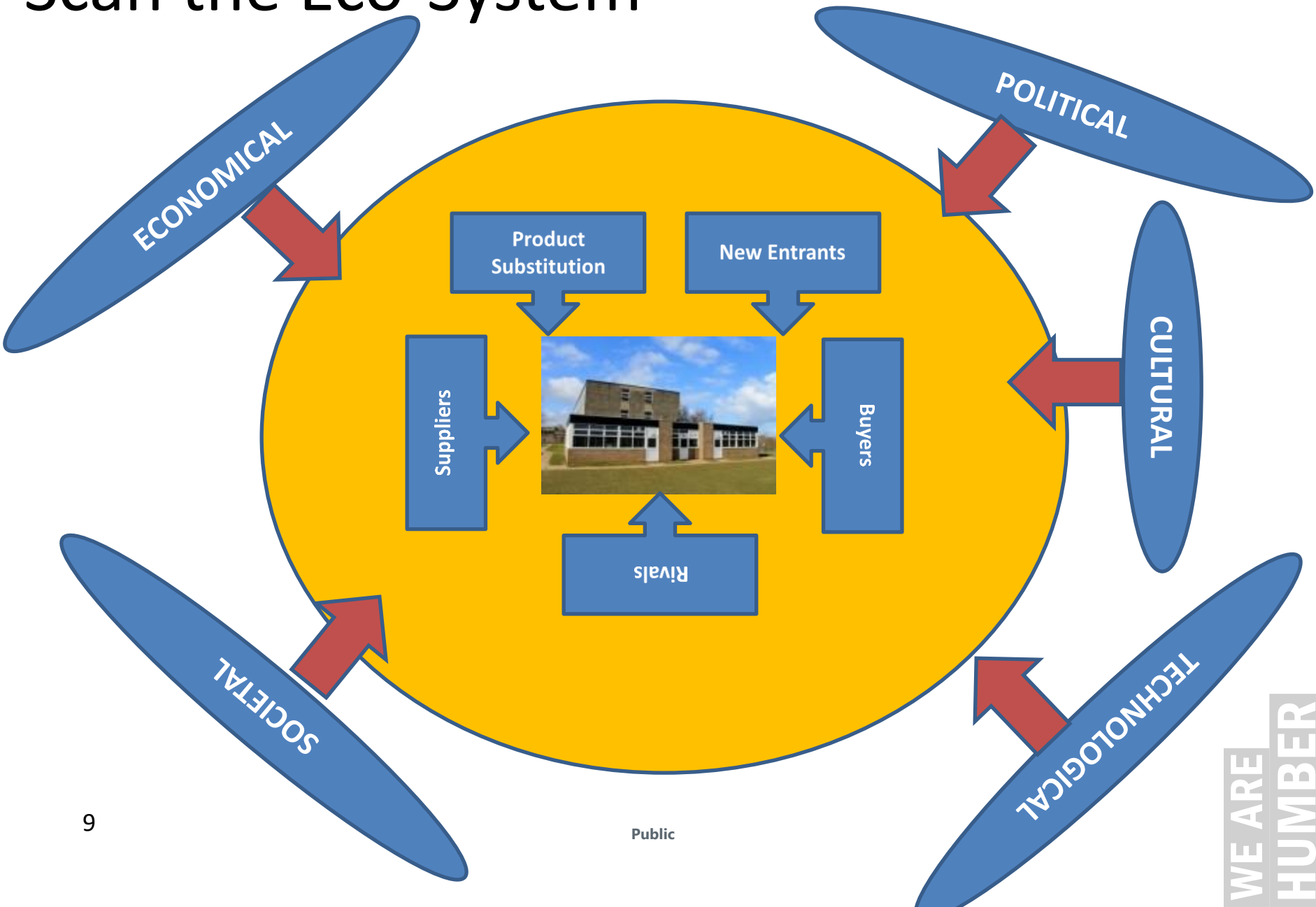
STO Model

- Project Charter should outline the success factors of the project during the initiation phase.

- Specifically, the charter should address if a particular project aligns with organization's financial strategy (e.g.: ROI)

$$\text{Return on Investment (ROI)} = \frac{\text{(Benefit – Cost)}}{\text{Cost}}$$

Scan the Eco-System



SWOT Analysis

Strengths Internal organizational strengths and can typically be controlled within the organization	**Weaknesses** Internal organizational weaknesses and can typically be controlled within the organization
Threats External factors that are threats and usually uncontrollable factors by the organization	**Opportunities** External factors that are opportunities and usually uncontrollable factors by the organization

Breakout Group Exercise 15-20 min

In your Breakout Group –

1) **Introduce yourselves to each other!**
2) **Identify and agree on a COMPANY that you all know about**
3) **List the various S, W, O, T of your chosen company**
4) **Come up with a possible VISION statement for your chosen company (no looking at their website for help!☺)**
5) **Nominate one person to share with rest of class**

Stakeholder Management

Stakeholder:
People, groups or organizations that could impact or be impacted by the project

Stakeholder Management:
A process to Identify stakeholders, analyze stakeholder expectations and their impact on the project, and develop appropriate management strategies for effectively involving stakeholders in project decisions and execution.

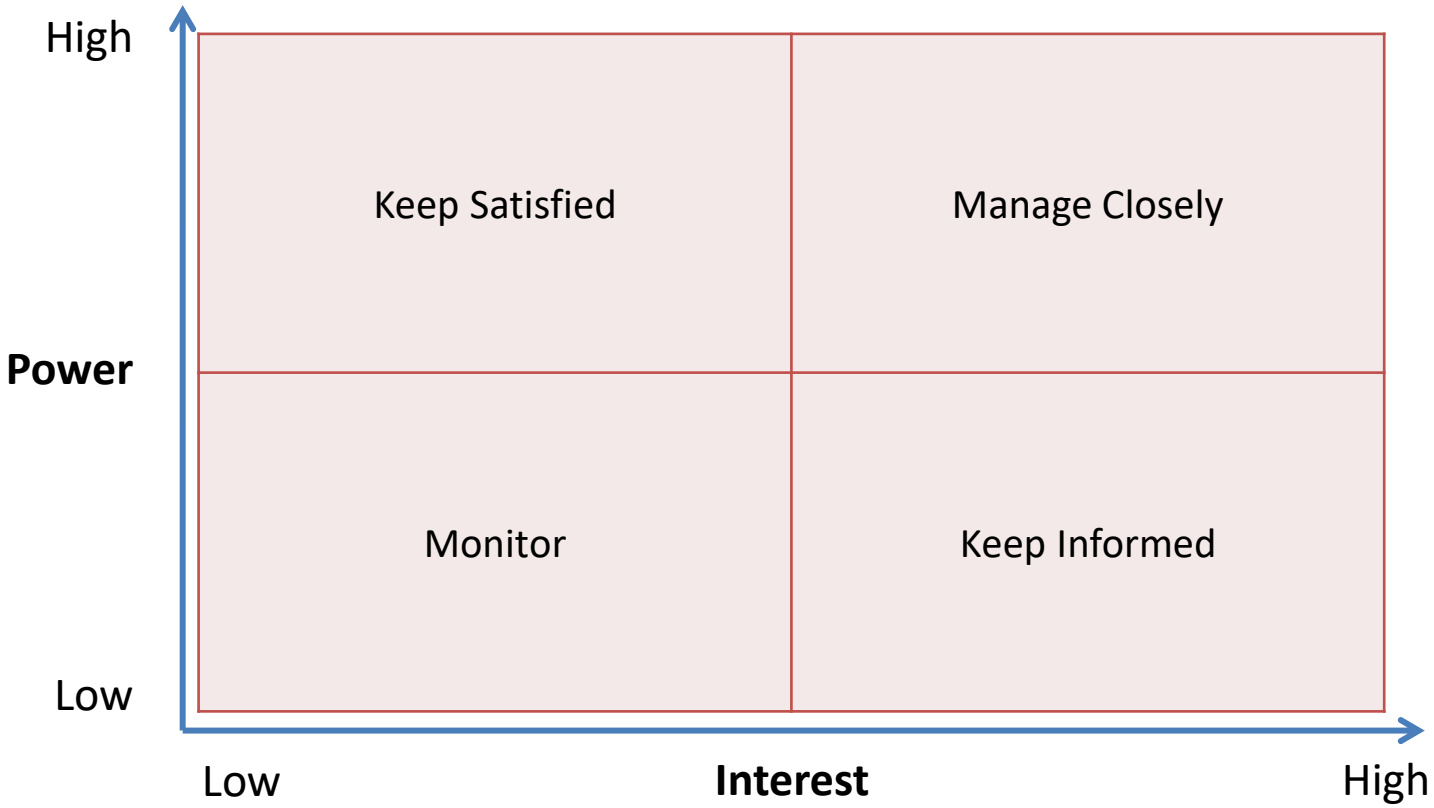
Stakeholder Register

- A table used to manage interactions with the stakeholders
- Lists all stakeholders and stakeholder groups
- Information added and updated throughout the phases of the project:
  - Interests, involvement, interdependencies, influence on project success
  - All interactions with each stakeholder or group, whether planned or not, whether initiated by the project or by the stakeholder
  - Who on the project team is responsible
- Used throughout the project
- Closely related to the project communication plan

Stakeholder Analysis

- Who are they?
- What are their interests?
- Will their interest level vary throughout the project?
- Can coalitions be built?
- The power/interest grid

Stakeholder power/interest grid



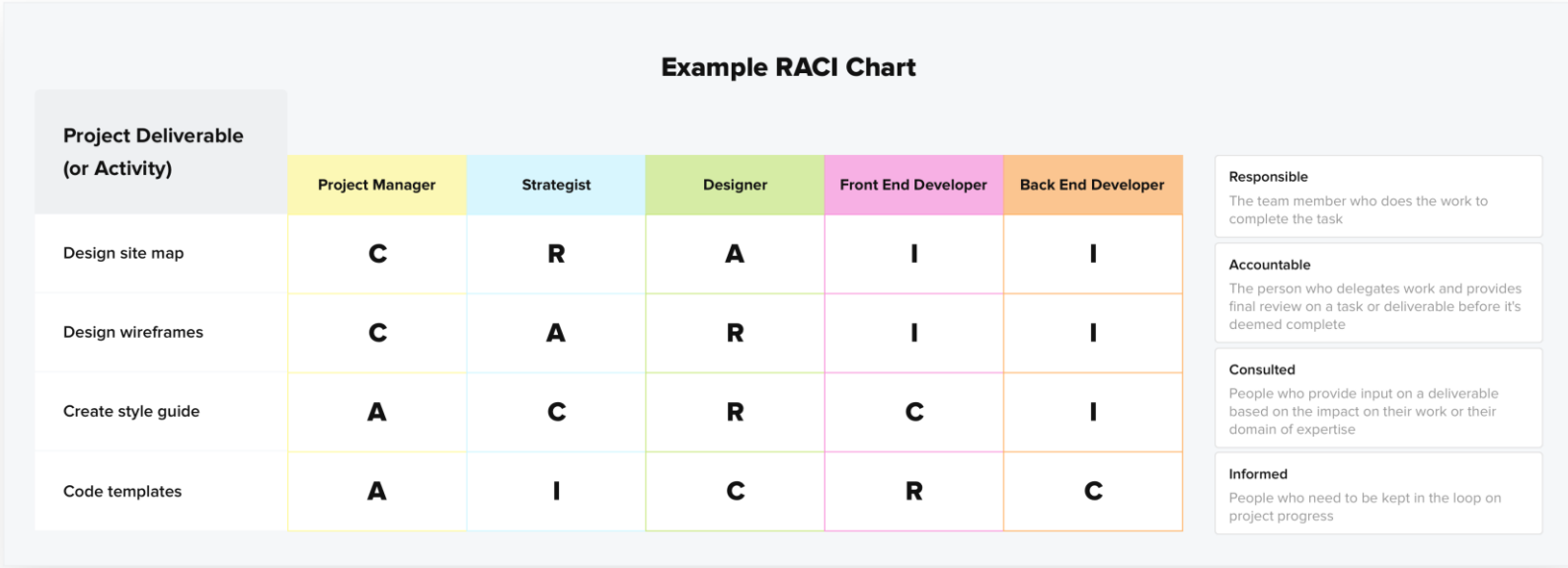
Identifying Roles and Responsibilities on projects

Some organizations use **RACI charts** to show Responsibility (who does the task), Accountability (who signs off on the task or has authority for it), Consultation (who has information necessary to complete the task), and Informed (who needs to be notified of task status/results) roles for project stakeholders.

▸A RACI chart lists tasks vertically, individuals or groups horizontally, and each intersecting cell contains an R, A, C, or I. Each task may have multiple R, C, or I entries, but there can be only one A entry to clarify which particular individual or group is accountable for each task.

▸For example, a mechanic is responsible for repairing a car, but the shop owner is accountable for the repairs getting done properly. Note that some people reverse the definitions of responsible and accountable.

RACI



Example RACI Chart

Project Deliverable (or Activity)	Project Manager	Strategist	Designer	Front End Developer	Back End Developer
Design site map	C	R	A	I	I
Design wireframes	C	A	R	I	I
Create style guide	A	C	R	C	I
Code templates	A	I	C	R	C

Responsible
The team member who does the work to complete the task

Accountable
The person who delegates work and provides final review on a task or deliverable before it's deemed complete

Consulted
People who provide input on a deliverable based on the impact on their work or their domain of expertise

Informed
People who need to be kept in the loop on project progress

A systems approach to managing projects

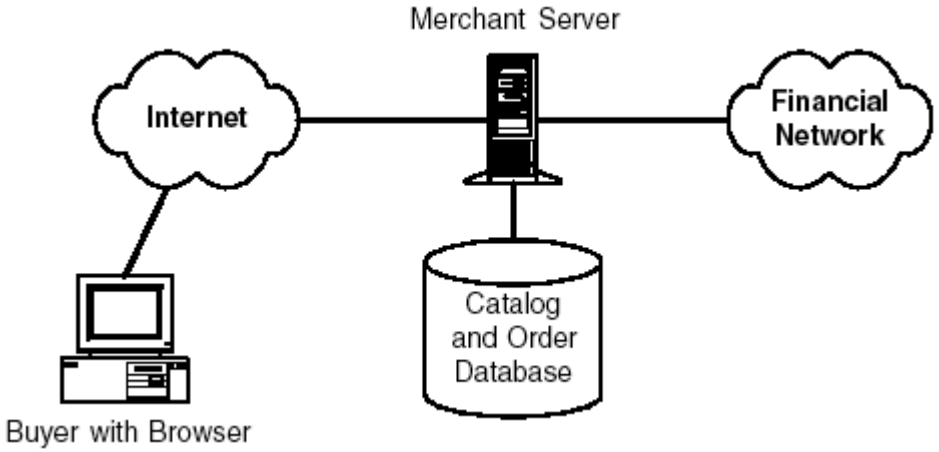
The term **systems approach** emerged in the 1950s to describe a holistic and analytical approach to solving complex problems that includes using a systems philosophy, systems analysis, and systems management.

- A **systems philosophy** is an overall model for thinking about things as systems. **Systems** are sets of interacting components working within an environment to fulfill some purpose. For example, the human body is a system composed of many subsystems the nervous system, the skeletal system, the circulatory system, the digestive system, and so on.
- **Systems analysis** is a problem-solving approach that requires defining the scope of the system, dividing it into its components, and then identifying and evaluating its problems, opportunities, constraints, and needs. Once this is completed, the systems analyst then examines alternative solutions for improving the current situation, identifies an optimum, or at least satisfactory, solution or action plan, and examines that plan against the entire system.
- **Systems management** addresses the business, technological, and organizational issues associated with creating, maintaining, and making changes to a system.

System Architecture

A **systems development life cycle (SDLC)** is a framework for describing the phases involved in developing information systems. Some popular models of an SDLC include the waterfall model, the iterative (spiral) model, the incremental build model, and the Agile model.

A system architecture is **the conceptual model that defines the structure, behavior, and more views of a system**. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



Details: Predictive (Waterfall)

Fully plan-driven approach where the three main project constraints (time, scope, cost) are all determined at a detailed level at the start of the project

Additional Details:

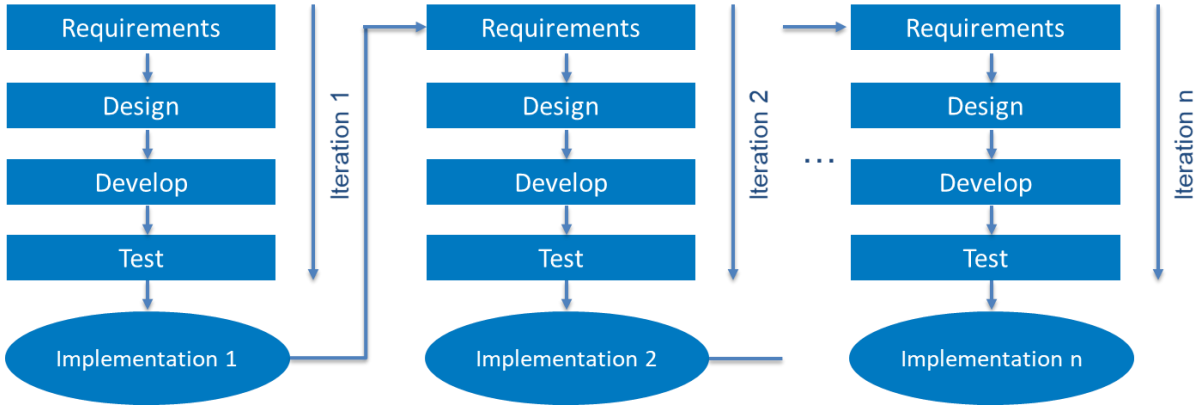
- You need to know your requirements going in
- The scope is fixed at the beginning
- There are distinct phases of activities: Requirements, Design, Development, Testing and Deployment
- A linear-sequential life cycle model; each phase starts after the previous phase completes
- The software is deployed into production after the completion of the Testing phase
- At the end of each phase, a review (or gate) takes place to determine if the project is on the right path and fit to move forward to the next phase of the project.

When to Use:

- Only when the requirements are very well known, clear and fixed
- Product definition is stable
- Technology is understood
- There are no ambiguous requirements
- Resources with required expertise are readily available

Details: Iterative

Breaking down the software development into smaller pieces, enabling better requirements definition at the start of each cycle.. Each release of is developed in a specific and fixed time period – called an “iteration”.



Additional Details:

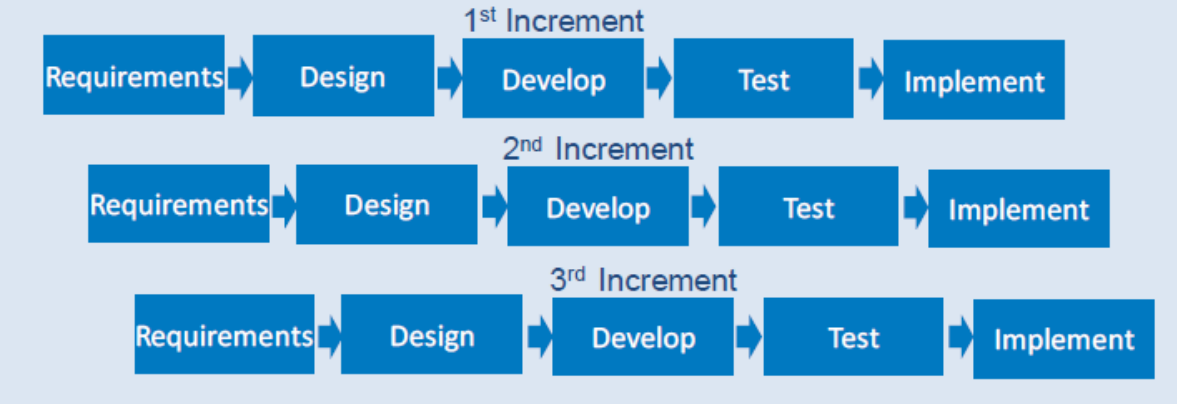
- There are the same five phases of activity
- Starts with a simple development of a small set of requirements, which iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed
- PMI (PMBOK Guide) recommends scope definition early in the project, but time and costs can be modified after each iteration when they are better understood
- This approach is comparable to many waterfall cycles with the customer verifying the work at the exit of each cycle
- It provides flexibility to address changes, which reduces risk
- The requirements are detailed for the next phase when you are done with the previous

When to Use:

- When requirements of the complete system are defined and understood
- Major requirements are defined, while some functionalities and requested enhancements evolve with the development process
- A new technology is being used and is being learned by the development team, while they are working on the project
- If there are some high-risk features and goals, which might change in the future

Details: Incremental

The incremental lifecycle approach develops a product through the implementation of incremental steps which have predetermined timeframes. Each increment delivers additional functionality for the product and is repeated until the final deliverable is produced.



Additional Details:

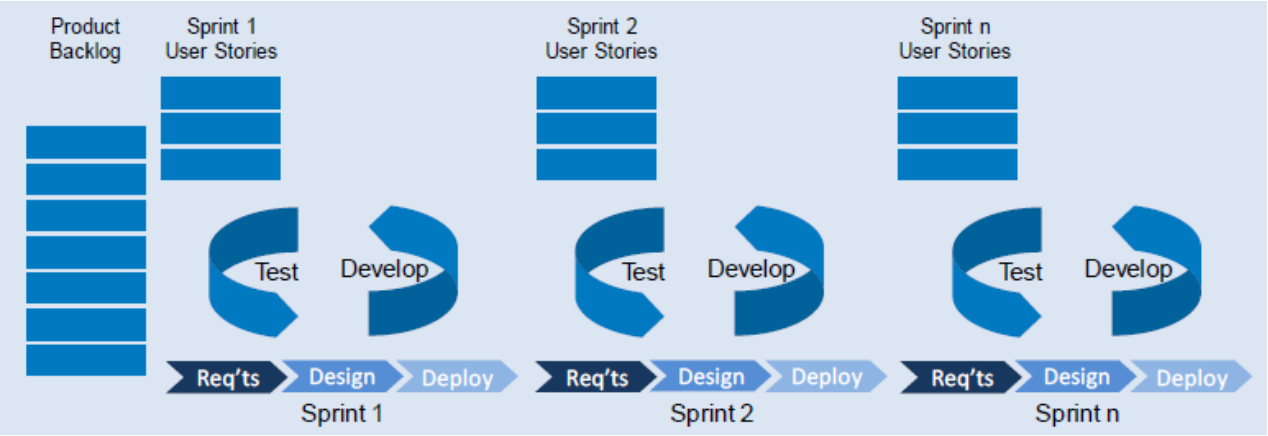
- Often, the Incremental approach is grouped with Iterative
- They are different but have some similarities
- Like with the Iterative approach, customers sign-off at each exit point

When to Use:

- When functionality can be split-up into increments (portions), whereby some functionality is delivered in each increment.
- When prototyping is a preferred approach

Details: Agile

The Agile Framework applies an incremental approach to work based on transparency, inspection, and adaptation, while also optimizing predictability and controlling risk.



Additional Details:

- Doesn't have separate phases of Requirements, Design, Development or Testing
- Combines both iterative and incremental approaches
- Requirements are represented by User Stories that are created and refined prior to being assigned to a Sprint; with the remaining User Stories sitting in the Backlog
- Design, development and testing are done together within Sprints
- Sprints generally last between two to four weeks each and include a limited number of User Stories
- There are many sprints within a product development release, depending upon the number of User Stories to complete

When to Use:

- Shorter planning based on iterations and multiple deliveries
- Flexible, cross-functional team composition
- Changes in deliverables are expected and less impactful
- Product delivered in functional stages
- Collaborative and interactive approach to requirements
- Customer is involved throughout the sprint
- Concurrent approach seeks to reduce dependencies

Summary of Characteristics by Framework

	Predictive (Waterfall)	Iterative/ Incremental	Agile
Requirements	Are complete, clearly defined, documented, and baselined in the planning phase of the project	Are for a set of features, rather than complete set of requirements for the project	Represented by the User Stories that are created and refined prior to being assigned to a specific sprint
Definition of Ready	Analysis & design is completed for all requirements before development begins	Analysis & design is completed for a 'set of requirements' before development begins	Requirements (user stories) that are not subject to change, that can be completed within the sprint will be considered for the sprint
Scoping	Sponsor (AE) decides project scope	Project Manager decides project scope for the iteration with Sponsor (AE)	Team decides the scope of the sprint based on the capacity & product backlog prioritization
Status Reports	Prescribed template (Clarity)		Daily Stand-up meeting (Scrum)
Task Assignment	PM assigns the tasks for the entire project		Team members are empowered to own the tasks for the entire sprint
Resource Utilization	Role specific; Resources only work on their assigned tasks	Role switchover is rare; Resources extend only on an 'as needed' basis	Role switchover is common; Everybody is ready to work on any task to complete sprint
Definition of Done	Delivery of artifacts that are developed, approved, and delivered at end of respective phase	Phase wise delivery of features (analysis & design in one phase, followed by development & testing in another phase) within the same iteration	A potentially shippable minimum viable product or the sprint
Quality Control	Detection and fix during the testing phase at the end of the project	Early detection and fix in each iteration followed by regression testing before deployment	Early detection and fix in each sprint
Delivery	Artifacts by phase and delivery of working software at end of project	Working software at end of each iteration	Working software at end of each sprint
Continuous Improvement (Lessons Learned)	At end of project; Lessons learned from previous projects implemented in next project	At end of every iteration; Lessons learned from previous iteration implemented in next iteration	Retrospective at end of every sprint; Lessons learned from previous sprint implemented in next sprint

Breakout Group Exercise 10-15min

In your Breakout Group –

1) **Discuss the pros and cons of waterfall vs agile frameworks**
2) **What type of projects would be better suited for waterfall, and which are better suited for agile?**
3) **Nominate one person to share with rest of class**

NEXT WEEK:

- Project Lifecyle (Waterfall)
- Gathering requirements
- Lab #1- Use Case Diagrams

THANK YOU.

HUMBER

WE ARE HUMBER