DES329 Midterm Summary

Systems Analysis and Design (SAD)
SIIT DE-ASD Y3T2/2021 – By Paphana Yiwsiw (@waterthatfrozen)

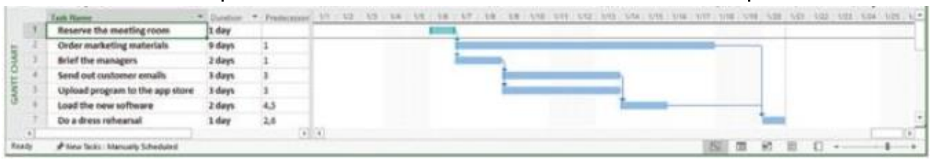
Lecture 3 - Managing System Projects

Overview of Project Management

- Project management: plan, schedule, monitor, control, and report on system development.
- Successful projects must be completed on time, in budget, meet requirements, and satisfy user.
- Project Triangle: Cost, Scope, Time; Any change in one leg will affect the others.
  - Challenge is to find the optimal balance among the factors
- Roles of Project Manager (PM):
  - Planning: identify all tasks and estimate completion time and cost.
  - Scheduling: create a task timetable to show tasks, dependencies, critical tasks
  - Monitoring: guide, supervise, coordinate team workload.
  - Reporting: create regular process reports for management, users, and project team.

Creating a Work Breakdown Structure (WBS)

- Breaking down a project into a series of smaller task
  - Gantt Chart: horizontal bar -> set of tasks, show plan and actual progress, simplify complex project using a task group
  - PERT/CPM Chart: Program Evaluation Review Technique/Critical Path Method; PERT utilizes a bottom-up technique, useful for scheduling, monitoring, controlling work, displaying complex task pattern and relationships
- Identify tasks in a WBS
  - Task: any work that has a beginning and end. requires the use of company resource, should be small and manageable.
  - Milestone: projects have milestone, recognizable reference points for monitor progress.
  - Listing tasks: task might be embedded in a document.
  - Duration estimation: can be hrs/d/wk. person-day: work that person complete in a day.
  - Time estimated by project managers: Best-case (O), Probable-case (r), Worst-case (p)
  - After estimation, the manager assign weight to each estimate -> calculate task duration
    - **Expected task duration (ET)** = $(o + 4r + p)/6$
  - Factors affecting duration
    - Project size: identify all task and time required. consider time taken for events affecting productivity.
    - Human resource: assemble team with skill and experience to handle the project and guide them. deal with factors that could affect the schedule.
    - Experience with similar project: develop time and cost estimates based on resources used by previous, similar IS projects.
    - Constraints: define requirements that can achieved within required constraints.



	Task Name	Duration	Predecessors
1	Reserve the meeting room	1 day	
2	Order marketing materials	9 days	1
3	Brief the managers	2 days	1
4	Send out customer emails	3 days	3
5	Upload program to the app store	3 days	3
6	Load the new software	2 days	4,5
7	Do a dress rehearsal	1 day	2,6

Task Patterns

- Arrangement of tasks in **LOGICAL SEQUENCE.**
    - Dependent tasks / Multiple successors tasks / Multiple predecessor tasks.
    - Using task boxes to create a model

Task Name	
Start Day	Task ID
Finish Day	Task Duration

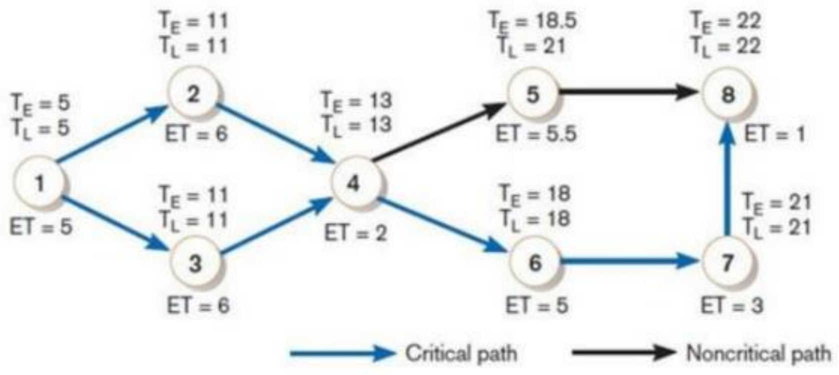
- Dependent task: completed in a sequence; one can initiated after prior has been completed.
- Multiple successor task: task that can initiated simultaneously, concurrent.
- Multiple predecessor task: task that can be initiated on completion on 2 or more prior tasks.
- Working with complex task pattern
    - When several task patterns combine, facts must be carefully studied to understand logic and sequence. A project schedule will not be accurate if underlying pattern is incorrect.

Critical Path

- Series of tasks which if delayed will affect the completion date of project.
    - If any task on the critical path falls behind schedule, the project is **delayed.**
- Calculate critical path
    - review patterns and determine start and finish dates, which will define critical path.
    - **Slack time** is the amount of time that task could be late without pushing back the completion date of the entire project.

Constructing a Gantt Chart and Network Diagram

1. Identify each activity to be completed in the project.
2. Determine time estimates and calculate expected completion time for each activity.
3. Determine the sequence of activity and precedence relationship among all activities by constructing Gantt chart and network diagram.
4. Determine the critical path. Network diagram shows dependencies.
   TE: Earliest expected completion time.
   TL: Latest expected completion time, without delaying the project.
    a. Calculate TE: sum the ET from left to right
        i. TE of last activity represents the complete time that project should take.
        ii. If 2 or more precede a task, the largest ET among those is used in calculate ET.
    b. Calculate TL: subtract ET for each from right to left, toward the start
    c. Calculate the **slack time** = TL - TE
    d. Find critical path: tasks with slack time = 0 are on critical path.



Project Monitoring and Control

- Monitoring and control techniques
    - Structured walk-through: review of a project member work by another team member.
        - Take place throughout SDLC.
        - Known as design, code, or testing review based on which phase they occur.
- Maintaining a schedule
    - Projects run into problems and delays
    - PM monitor and control work
        - Anticipate problems, avoid them, and minimize impact
        - Identify potential solution and select the best way to solve the problem.
- Tasks and the critical path
    - PM spends most of their time tracking the tasks along the critical path.

Reporting

- Project status meetings
    - PM schedule regular meetings: share updates, discuss problems, explain new techniques, help collect data
- Project status reports
    - PM communicates to supervisors, upper management, or users.
- Dealing with problems: decide how to handle problems can be difficult.

Project Management Software

- PM uses software applications to help plan, schedule, monitor, and report a project.
- Most programs offer features such as PERT/CRM, Gantt chart, etc.

Risk Management

- Steps in risk management
    - Develop a risk management plan
    - Identify the risks
    - Analyze the risks: Qualitative and Quantitative.
- Risk Matrix: Risk Impact x Risk Probability

Managing for Success

- Project management is a challenging task
- PM must be alert, technically competent, resourceful
- Projects get derailed for various reasons: business/budget/schedule issue.

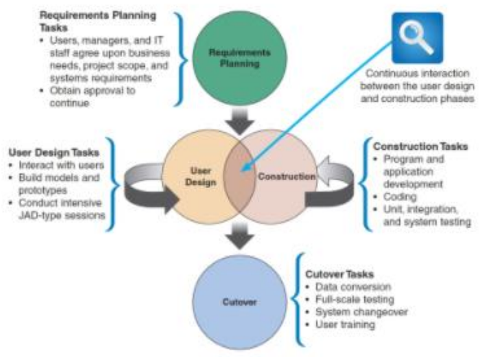
Lecture 4 - Requirements Engineering

System Requirements

- Feature that must be included in an IS to satisfy business requirements, benchmarks to measure overall acceptability.
- Requirements engineering activities includes gathering, representing, validating and verifying.
- Types of requirements:
    - Functional requirements: what services is provided by the system
      ex. the website reports online volume stats hourly and during peak periods, each input form must include date, time, product code, customer number, quantity.
    - Non-functional requirements: constraints of operational system, aka. quality attributes.
      ex. the system must support 25 users simultaneously. RT must not exceed 4 seconds.
- Requirements challenges
    - Imprecision: represented using natural language.
    - Agreement: SA and client agree on whether specific requirement has been met or not?
    - Creep: change requirements rapidly can cause problems for SA and team members.
- Additional consideration
    - Scalability: ability to handle increasing number of volume and transaction
    - Security: make system harder to infiltrate/penetrate
    - TOC, total cost of ownership: direct and indirect cost

Team-based Techniques

- **JAD: Joint Application Development**
    - Bring users into the development process as active participants
    - User involvement: active role, formal or informal
    - JAD participants and roles: project leader and one or more members.
        - ex. top management, managers, users, SA, developer, recorder.
    - Advantages: allow keys user to participate effectively, more likely to feel sense of ownership, produce more accurate system requirement, better common understanding of goals, stronger commitment to the new system.
    - Disadvantages: expensive than traditional method, cumbersome if group is too large.
- RAD: Rapid Application Development
    - Group approach like JAD
    - Product: new IS
    - Methodology: 4 phases life cycle, parallels the traditional SDLC. Reduce cost and development time / Increase the probability of success / Rely on prototyping and user involvement


Requirements Planning Tasks
• Users, managers, and IT staff agree upon business needs, project scope, and systems requirements
• Obtain approval to continue
Requirements Planning
Continuous interaction between the user design and construction phases
User Design Tasks
• Interact with users
• Build models and prototypes
• Conduct intensive JAD-type sessions
User Design
Construction
Construction Tasks
• Program and application development
• Coding
• Unit, integration, and system testing
Cutover
Cutover Tasks
• Data conversion
• Full-scale testing
• System changeover
• User training

    - Objectives: cut development time and expense, involve users in every phase.
    - Advantages: help system develops quickly with cost savings.
    - Disadvantages: doesn't emphasize strategic business needs & less time to develop quality, consistency, and design standards.
- **Agile Methods**
    - Attempt to develop a system incrementally by <u>building a series of prototypes and adjusting to the user requirements.</u>
    - Developers revise, extend, merge earlier versions to final products.
    - Emphasize continuous feedback
    - Advantages: flexible and efficient in dealing with change, frequent deliverables constantly validate the project, reduce risks.
    - Disadvantages: need high level of technical and interpersonal skills, lack of structure and documentation can introduce risk factors, subject to significant change in scope.
- **Scrum**
    - Scrum session has a specific guideline on emphasize time blocks, interaction, and team-based activities that resulted in deliverable software.

Gathering Requirements

- First step in requirements engineering process
- requirement elicitation or fact finding: who/what/where/when/how, about the procedure.
- Gathering requirement through interviews:
    - Interview process:
        - determine people to interview: select right people and ask right questions, consider from both formal and informal structure, also decides whether it is a group or individual interview.
        - establish objectives: determine the area to be discussed, list facts that need to be gathered, objectives depend on roles of the person being interviewed.
        - develop questions: decide what to ask and how to phrase it. avoid leading question, open-ended question gives spontaneous and unstructured responses, while close-ended limited response but useful when specific info is needed. Range-of-response use when evaluate something by providing limited answer to specific response on numeric scale.
        - prepare the interview: careful preparation is essential. no more than 1 hour, verify time, place, length, and topics. If questions about docs, samples are needed.
        - conduct the interview: develop a specific plan for the meeting.
          introduce yourself -> describe the project -> explain your objective.
          practice engaged listening, look for verbal and nonverbal response, allow person to have time to think about the question and answer. After the interview, summarize the session and seek a confirmation.
        - document the interview: minimize note taking, record information quickly after interview, send memo expressing your appreciation.
        - evaluate the interview: recording the fact obtained from the interview, identify any possible bias.
- Gathering Requirements using other techniques
    - Document review: review current documentation

- Observation: provide additional perspective and better understanding of system procedure, should plan in advance.
- Questionnaires and surveys: make sure questions collect the right data that can be used further.
- Brainstorming: small group discussion on specific problem
- Sampling: ensure the representation of overall population accurately, could be systematic (from every nth people), stratified (from each area), random sample.
- Research: obtain background info, technical material, trends, and developments.

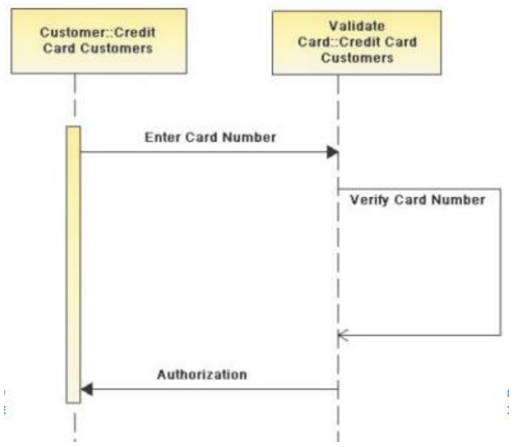
- Comparison between interviews and surveys
  - Interview is more familiar and personal but time-consuming and costly
  - Survey gives more opportunity to provide input and suggestion

Gathering Requirement in Agile Projects

- Agile method used for requirements gathering
  - Variation on interviews focused on features, user stories, and storyboarding.
  - Requirements are gathered and successively refined

Representing Requirements

- Principles for documentation: record as soon as obtained, simplest method, understandable by anybody else, organize to locate easily.
- Representation
  - Natural Language
    - Majority of requirement are represented using unstructured language
    - Problem with imprecision and lack of understanding
    - Can store on index card if represented as structured natural language.
  - Diagrams
    - Graphical methods and nontechnical languages represent stages of system
    - Functional decomposition diagram: top-down process representation
    - Business process diagrams: represent one or more business processes
      - BPMN is used to represent event, process, and workflow.
    - Data Flow Diagram (DFD): show how system store, process, and transform data.
    - Models: Unified Modeling Language (UML): Provide a more formal representation of system requirements, mostly using in object-oriented one, classes and interfaces.
    - Use Case Diagram: Visually represents the interaction between users and IS
    - Sequence Diagram: Show timing of interactions between objects


Customer::Credit Card Customers
Validate Card::Credit Card Customers
Enter Card Number
Verify Card Number
Authorization

Validating and Verifying Requirements (V&V)

- Concerned with demonstrate that requirements define the system that customer really wants.
- Validation: are correct requirements stated?
- Verification: are requirements stated correctly?
- Requirement attributes to be checks:
    - Validity: system provide functions to supports customer's need?
    - Consistency: any conflicting requirements?
    - Completeness: all required customer functions included?
    - Realism: requirements can implement in given budget and technology?
    - Verifiability: can requirement be checked?
    - Comprehensibility: origin of requirement stated clearly?
    - Adaptability: can requirement changes without large impact on other requirements?

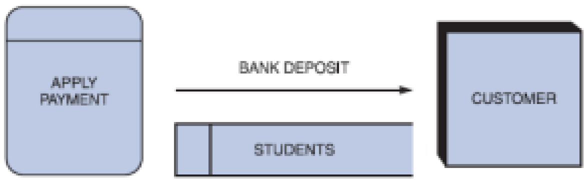
Lecture 5 - Data and Process Modeling

Logical VS Physical Models

- Logical model: show what system must do.
- Physical model: describe how system will be constructed.
- 4-model approach
    - Logical & Physical model of the current system
    - Logical & Physical model of the new system

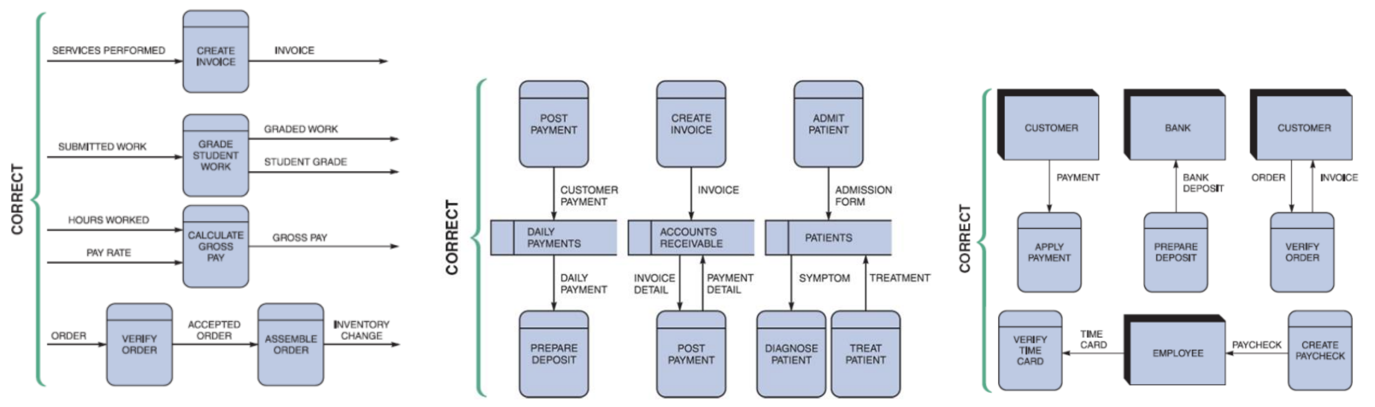
Data Flow Diagrams

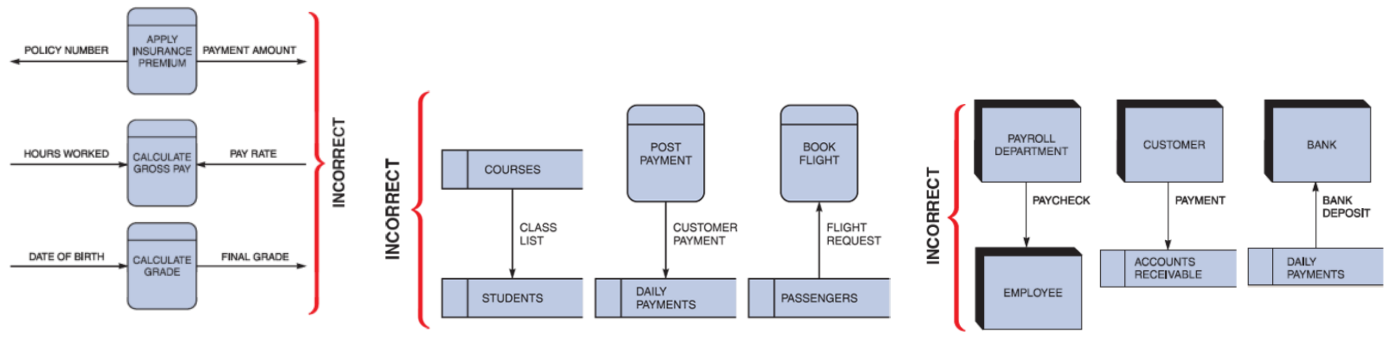
- uses symbols to show how the system transforms input data into useful information, how data moves through IS but not its logic or processing steps.
- 4 symbols of DFD

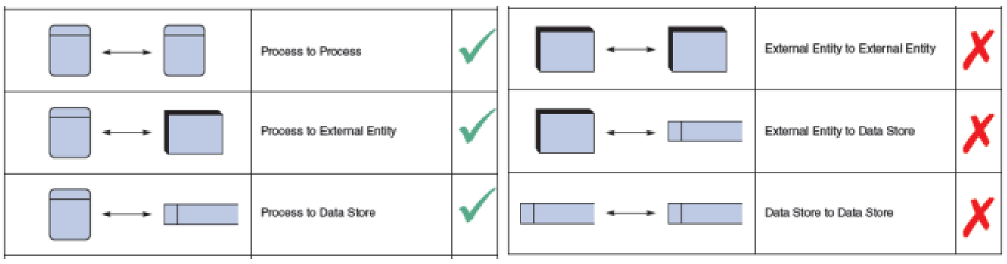


    - Process: receives input data and produces output.
        - Inside contains business logic that transform the data
        - Process name identifies a specific function
        - can be referred as a black box
    - Data Flow: show the flow of data
        - can be a single or double arrowhead line.
    - Data Store: represent data that system stores.
        - doesn't show the detailed contents of data stores, structure and data elements are defined in data dictionary.
        - must be connected to a process with data flow.
    - Entity: provide data to the system or receive output
        - can be referred as terminators because it is the origin or last destination of data.
- **Incorrect** usages of DFD symbols
    - Combination of data flow and process combination should avoid spontaneous generation (process generate output without any input), black holes (data flows into process without generate output), and grey hole (nonsense input produces output).
    - Data cannot flow between 2 data stores directly, and each data store should have an input and output data flow.
    - External entities must be connected to the process, cannot connected to another entity or data stores by itself.
- **Drawing Data Flow Diagrams**
    - created based on fact-finding results
    - Guidelines of creating DFDs
        - Draw the Context Diagram (CD: Level 0 DFD), so it fits on 1 page.
        - Use the name of the IS as the process name in the CD
        - Use unique names within each set of symbols
        - **Do not cross lines**

- Provide a unique name and reference number for each process
    - 0 for CD / 1, 2, 3, … for Level 1 DFD / use decimal places like 1.1, 1.2, … for lower
- Ensure that the model is accurate, easy to understand, and meet user needs.



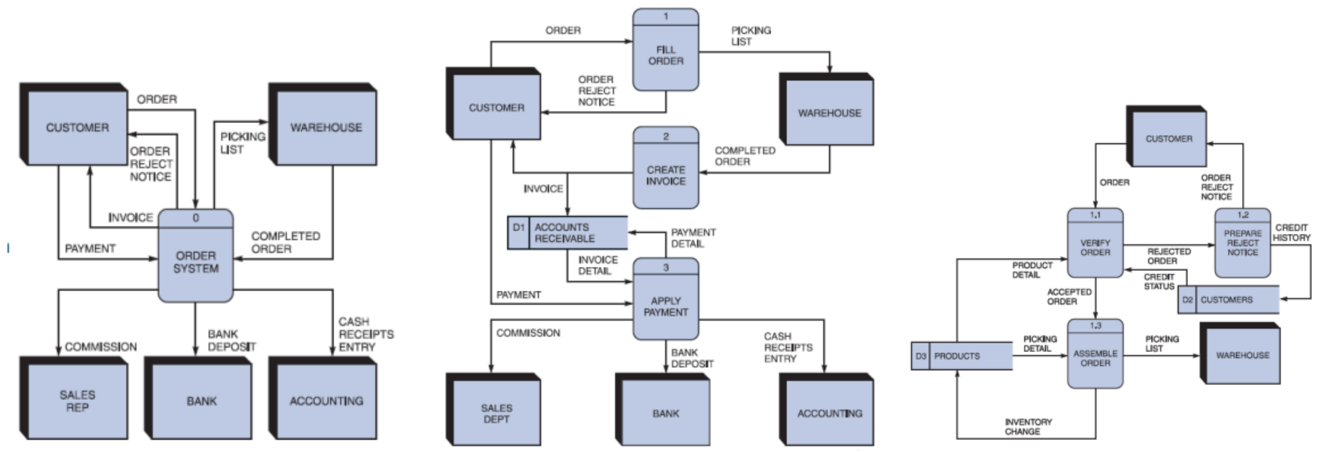




	Process to Process	✓		External Entity to External Entity	✗
	Process to External Entity	✓		External Entity to Data Store	✗
	Process to Data Store	✓		Data Store to Data Store	✗

Context Diagram & Level-0 Diagram

- Context diagram (CD) is an overview of IS system that shows system boundaries, external entities that interact with the system, major information flows between entities and system.
- Level-0 diagram is a DFD that represents major processes of the system, data flows, and data stores at high level of detail.
- Lower-level DFDs are the more primitive DFD of each major process, labelled as 1.1, 1.2, 1.3, …



------------------------------------ **Not included in midterm exam** ------------------------------------

Data Dictionary

- Central storehouse of information about a system data
- **Define and describe all data elements and combinations of data elements**
- SA uses the data dictionary to collect, document, organize specific facts about a system.
- Data elements (aka. data item or field) is the smallest piece of data that has meaning within IS. Combined into records (aka. data structure)
  - Record is a combination of data elements that included in data flow or kept in data store.
- Every data element in data dictionary should be documented , provide clear & comprehensive information about the data and processes that make up a system.
- Record and describe attributes: data element name and label, alias, type, length, default value, acceptable value, source, security, responsible user, description, and comments.
- Documenting data flows: data flow name, label, description, alternate name, origin, destination, record, volume, and frequency.
- Documenting data stores: data store name, label, description, alternate name, attributes, volume and frequency.
- Documenting processes: process name, label, description, and process number.
- Documenting entities: entity name, description, alternate name, input and output data flows.
- Documenting records: data structure name, definition, description, alternate name, attributes.
- Data dictionary reports
  - Alphabetized list of all data elements
  - Report describing each element and indicate user or department.
  - Report of all data flows and data stores that use a particular data element.
  - Detailed reports showing all characteristics of data elements, records, flows, processes, and other item stored in the data.

Process Description Tools in Modular Design

- Document details of primitives and represents a set of processing steps and business logic.
- Process descriptions in O-O development
  - O-O combines data and processes that act on data into objects, and similar objects can be grouped together into classes. (O-O processes -> methods)
- Modular design
  - Based on combinations of logical structures, served as building blocks for process
    - Sequence / Selection / Iteration
- Structured English
  - Describe logical processes clearly and accurately
  - use only the three building blocks of sequence, selection, and iteration.
  - use indentation for readability, and a limited vocabulary
  - sort of like pseudocode.
- Decision tables
  - logical structure that shows every combination of conditions and outcomes.
  - number of rules doubles each time a condition is added.
  - best way to describe complex set of conditions
  - sort of like a truth table in digital circuit.
- Decision trees
  - Graphical representation of conditions, actions, and rules in a decision table.