SOFTWARE REQUIREMENT ENGINEERING

LECTURE NO: 12

BY: NAZISH NOUMAN

A picture is worth 1024 words

Modeling the requirements

Visual requirements models can help you identify missing, extraneous, and inconsistent requirements. It includes:

- Dataflow diagrams
- Process flow diagrams such as swimlane diagrams
- State-transition diagrams (STDs) and state tables
- Dialog maps
- Decision tables and decision trees
- Event-response tables
- Feature trees
- Use case diagrams
- Activity diagrams
- Entity-relationship diagrams (ERDs)

From voice of the customer to analysis models

By listening carefully to how customers present their requirements, the business analyst can pick out keywords that translate into specific model elements

Type of word	Examples	Analysis model components
Noun	People, organizations, software systems, data elements, or objects that exist	■ External entities, data stores, or data flow (DFD) ■ Actors (use case diagram) ■ Entities or their attributes (ERD) ■ Lanes (swimlane diagram) ■ Objects with states (STD)
Verb	Actions, things a user or system can do, or events that can take place	■ Processes (DFD) ■ Process steps (swimlane diagram) ■ Use cases (use case diagram) ■ Relationships (ERD) ■ Transitions (STD) ■ Activities (activity diagram) ■ Events (event-response table)
Conditional	Conditional logic statements, such as if/then	■ Decisions (decision tree, decision table, or activity diagram) ■ Branching (swimlane diagram or activity diagram)

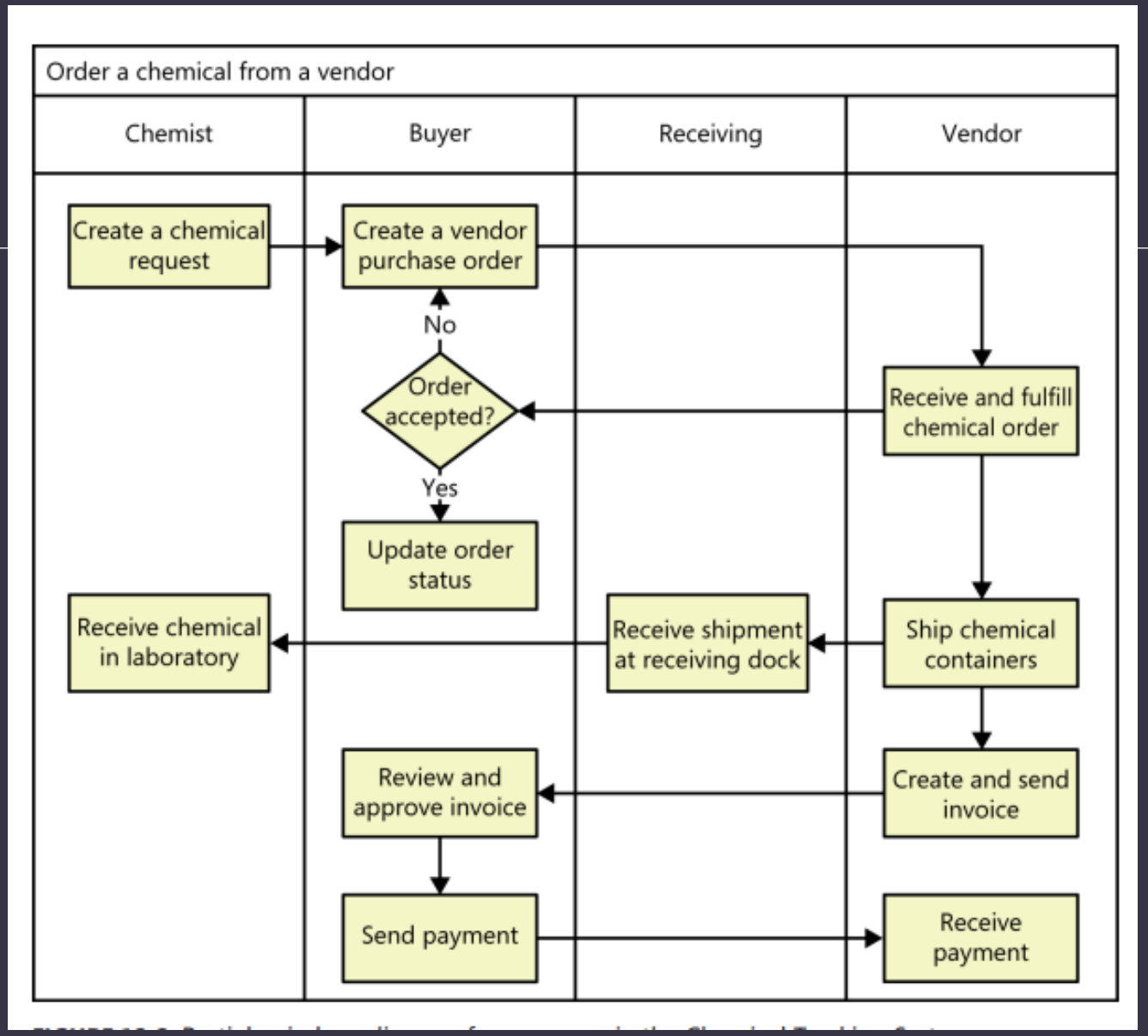
Nouns are highlighted in **bold**, verbs are in *italics*, and conditional statements are in ***bold italics;***

A **chemist** or a member of the **chemical stockroom staff** can *place* a **request** for one or more **chemicals** ***if the user is an authorized requester.*** **The request can be *fulfiled* either by *delivering* a container of the chemical that is already in the chemical stockroom's inventory or by placing an order for a new container of the chemical with an outside vendor*. If the chemical is hazardous,* the chemical can be delivered only *if the user is trained.* The person placing the request must be able to *search* vendor catalogs online for specifi chemicals while *preparing* his request. The system needs to *track* the status of every chemical request from the time it is prepared until the request is either fulfiled or *canceled.* It also needs to track the history of every chemical container from the time it is *received* at the company until it is fully *consumed* or *disposed* of.**

Swimlane diagram

Swimlane diagrams* provide a way to represent the steps involved in a business process or the operations of a proposed software system. They are a variation of flowcharts, subdivided into visual subcomponents called *lanes.

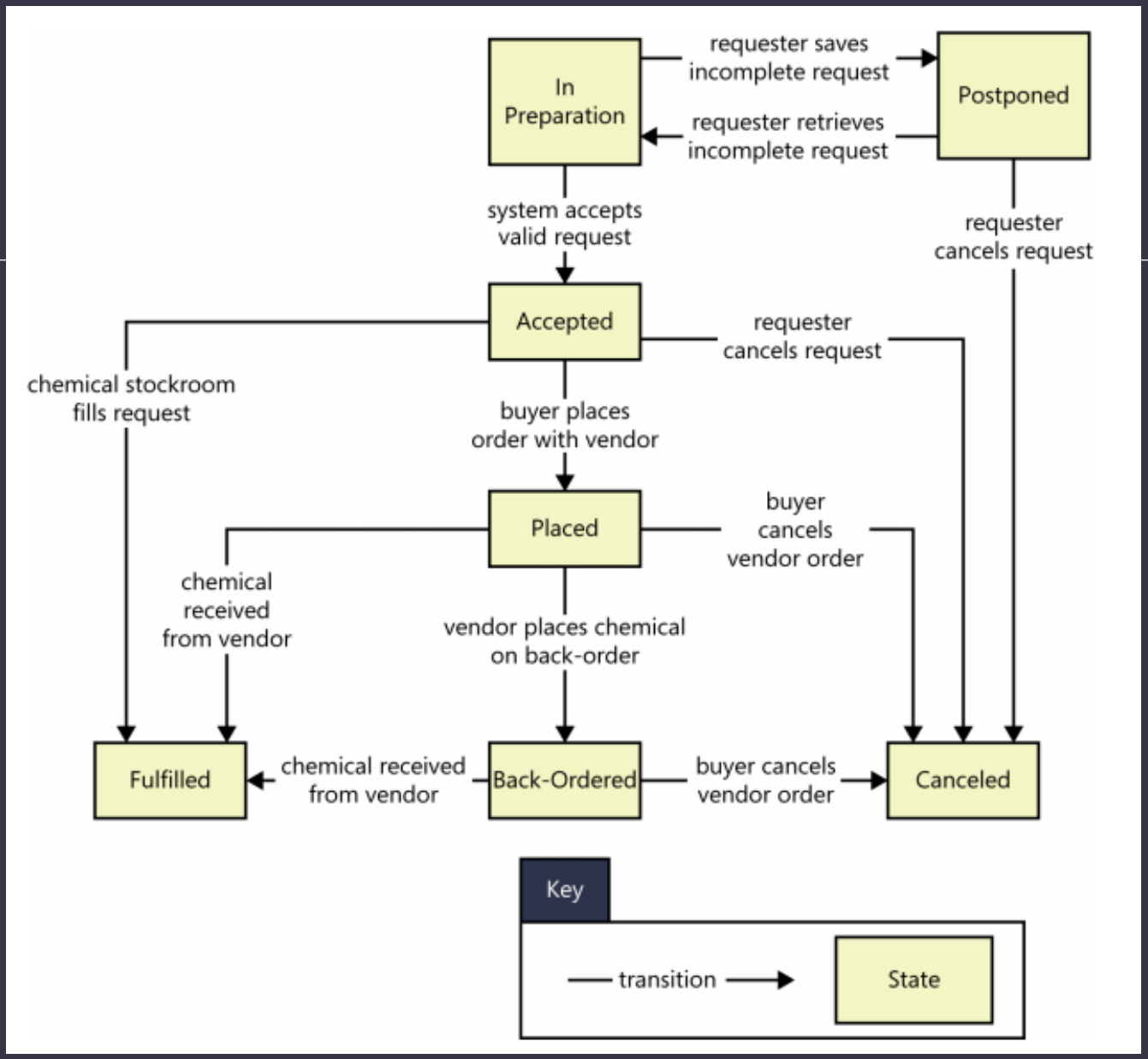
- Swimlane diagrams are sometimes called *cross-functional diagrams.* They are similar to UML activity diagrams.
- The swimlane diagram is one of the easiest models for stakeholders to understand because the notation is simple and commonly used.
- Drafting business processes in swimlane diagrams can be a good starting point for elicitation conversations
  - Process steps, shown as rectangles.
  - Transitions between process steps, shown as arrows connecting pairs of rectangles.
  - Decisions, shown as diamonds with multiple branches leaving each diamond. The decision choices are shown as text labels on each arrow leaving a diamond.
  - The lanes are most commonly roles, departments, or systems. They show who or what is executing the steps in a given lane.


Order a chemical from a vendor
Chemist
Buyer
Receiving
Vendor
Create a chemical request
Create a vendor purchase order
No
Order accepted?
Yes
Update order status
Receive and fulfill chemical order
Receive chemical in laboratory
Receive shipment at receiving dock
Ship chemical containers
Review and approve invoice
Create and send invoice
Send payment
Receive payment

State-transition diagram and state table

- Software systems involve a combination of functional behavior, data manipulation, and state changes.
- Real-time systems and process control applications can exist in one of a limited number of states at any given time. A state change can take place only when well-defied criteria are satisfied.

State-transition diagrams **and** ***state tables*** **are two state models that provide a concise, complete, and unambiguous representation of the states of an object or system. The state-transition diagram (STD) shows the possible transitions between states visually.**

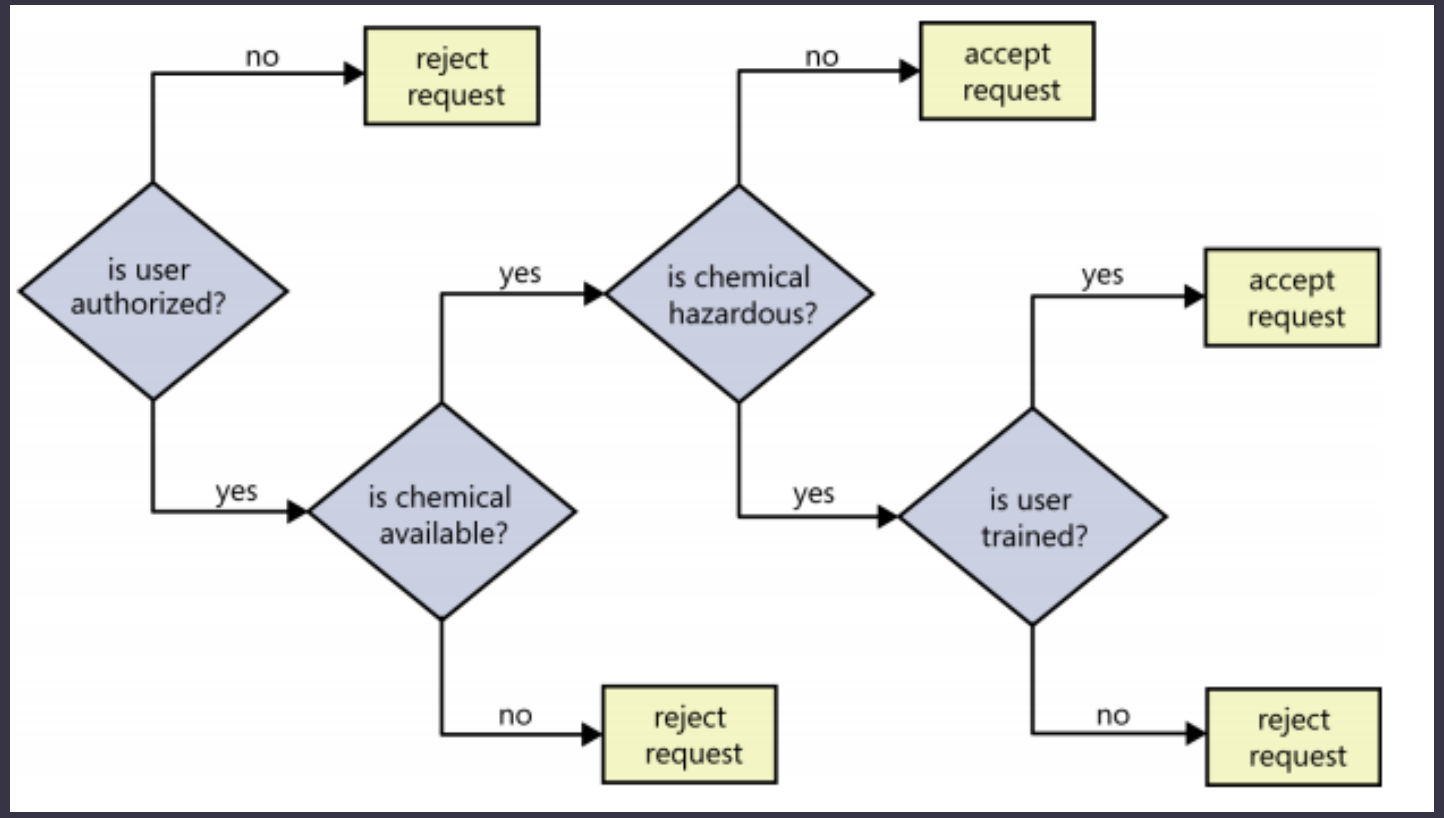

In Preparation
requester saves incomplete request
Postponed
requester retrieves incomplete request
system accepts valid request
requester cancels request
Accepted
requester cancels request
chemical stockroom fills request
buyer places order with vendor
Placed
buyer cancels vendor order
chemical received from vendor
vendor places chemical on back-order
Fulfilled
chemical received from vendor
Back-Ordered
buyer cancels vendor order
Canceled
Key
transition
State

This STD shows that an individual request can take on one of the following seven possible states:

- **In Preparation** The Requester is creating a new request, having initiated that function from some other part of the system.
- **Postponed** The Requester saved a partial request for future completion without either submitting the request to the system or canceling the request operation.
- **Accepted** The Requester submitted a completed chemical request and the system accepted it for processing.
- **Placed** The request must be satisfied by an outside vendor and a buyer has placed an order with the vendor.
- **Fulfilled** The request has been satisfied, either by the delivery of a chemical container from the chemical stockroom to the Requester or by receipt of a chemical from a vendor.
- **Back-ordered** The vendor didn't have the chemical available and notified the buyer that it was back-ordered for future delivery.
- **Canceled** The Requester canceled an accepted request before it was fulfilled, or the buyer canceled a vendor order before it was fulfilled or while it was back-ordered.

Decision trees

Decision trees are useful ways to document requirements (or business rules) to avoid overlooking any combinations of conditions.



A state table shows all of the possible transitions between states in the form of a matrix.

	In preparation	Postponed	Accepted	Placed	Back-Ordered	Fulfilled	Canceled
In Preparation	no	user saves incomplete request	system accepts valid request	no	no	no	no
Postponed	user retrieves incomplete request	no	no	no	no	no	no
Accepted	no	no	no	buyer places order with vendor	no	chemical stockroom fills request	requester cancels request
Placed	no	no	no	no	vendor places chemical on back-order	chemical received from vendor	buyer cancels vendor order
Back-Ordered	no	no	no	no	no	chemical received from vendor	buyer cancels vendor order
Fulfilled	no	no	no	no	no	no	no
Canceled	no	no	no	no	no	no	no

FIGURE 12-4 State table for a chemical request in the Chemical Tracking System.

Event-response tables

An *event* is some change or activity that takes place in the user's environment that stimulates a response from the software system (Wiley 2000).

An *event-response table* (also called an *event table* or an *event list*) itemizes all such events and the behavior the system is expected to exhibit in reaction to each event.

Event Classes

There are three classes of system events

Business event A business event is an action by a human user that stimulates a dialog with the software, as when the user initiates a use case. The event-response sequences correspond to the steps in a use case or swimlane diagram.

Signal event A signal event is registered when the system receives a control signal, data reading, or interrupt from an external hardware device or another software system, such as when a switch closes, a voltage changes, another application requests a service, or a user swipes his finger on a tablet's screen.

Temporal event A temporal event is time-triggered, as when the computer's clock reaches a specified time (say, to launch an automatic data export operation at midnight) or when a preset duration has passed since a previous event (as in a system that logs the temperature read by a sensor every 10 seconds).

Systems respond to business, signal, and temporal events