Monday June 20, 2016

Requirements Analysis Models

* Different views/perspectives of requirements artifacts
* Benefits include:
  + Can give a clear picture of what customer wants which helps find gaps, conflicts, inconsistencies in requirements
* Provides a bridge to design phase
* Can make it easier for customer to validate

Types of Analysis Models

* **Scenario** – use cases and diagram, user stories, usages scenarios, and sequence diagrams
* **Structural** – object details, class diagrams, ERDs
* **Behavioral** – state diagrams
* **Flow** – DFDs, activity diagrams

The UML

* Unified Modeling Language
* Developed by Rational Software Corp. in mid 90’s
* A standardization of software modeling notation
  + Adapt, simplify, expand, merge widely used notations of the day
* Widely use today (at least some parts of the UML)

Modeling Guidelines

* Stay high-level and abstract
* Don’t get technical unless client requires it
* Loose coupling
* Models should be useful for all Stakeholders
* Keep it simple (complex models should be decomposed)

Use Case Diagrams

* Simple model for analyzing which actors (humans and external systems) can access what functionality.
* Can be used to guide further requirements elaboration and for access control testing
  + Can take the place of a list of user roles and use cases

Activity Diagrams

* UML flow model for things like
  + Use case main and alternate flows
  + Operations/Functions
  + Objects/Classes
  + Business processes
  + Anything that has a flow/sequence of transformation
* Node Types
  + **Action** **node** – performs work; all inputs required for node to generate output
  + **Decision** **node** – branches based on guard condition
  + **Merge** **node** – merges multiple paths together; any input causes merge node to output
  + **Initial** **node** – flow starts here
  + **Final node** – flow ends here
  + **Guard condition** – condition that affects decision branch
  + **Decision condition** – more complex guard condition.
  + **Receive Signal** – responds to an external signal and initiates flow
  + **Time Event** – generates flow at specified moment, interval, or period
  + **Send Signal** – sends a signal to a receiving Activity Diagram

Activity Diagram Swimlanes (Partitions)

* Group related actions by
  + Use case (diagram shows interaction between use cases)
  + Classes (data objects)
  + Entity/User Role
  + Business units (i.e., Production and Warehouse)

Objects in Activity Diagrams

* Represent instances of an object as it is transformed
* Object flow is object’s movement through the model
* Objects are created and consumed by action nodes
* Can show state or buffer semantics with Object
* Can be inputs or output of activity diagram

Sequence Diagrams

* Show time-based sequence of events between entities and data objects
* Provide more detailed view of a use case scenario
* Lifeline – represents a participant in an interaction
  + i.e. an actor, class/object, subsystem
* Similar to activity diagrams with swimlanes