

USING DRIVER-BASED DATA TO CREATE AN IES MODEL

Specifically, to create IES's cost functions

1. Assign all cost objects to one of IES's six types of cost objects
2. Determine cost drivers for cost objects
3. Separate fixed costs from variable costs and assign them to appropriate cost objects
4. Compute slope of variable portion of cost functions and step costs
 - $\text{Slope} = \$/\text{driver} = \text{activity}/\text{driver} \times \text{resource}/\text{driver} \times \$/\text{resource}$
 - If set up/change over or maintenance involved, increase slope appropriately
5. Add capacity constraints to cost functions as required
 - If set up/change over or maintenance involved, decrease capacity appropriately
6. Create model structure

USING DRIVER-BASED DATA IN AN IES MODEL

Example: CAM -I's *The Closed Loop*

1. Define cost objects
 - Products = support for a campaign, a campaign
 - Activities = support a campaign, execute a campaign
 - Labor = reps, supervisors, general manager
 - Facility = call center
 - Support = telcom, sundry, lease
2. Assign all cost objects to one of IES's six types of cost objects where not obvious
 - Lease = facility
 - Reps, telcon = campaign product
 - Sundry, supervisors, general manager = support product
3. Determine cost drivers for cost objects
 - Cost driver = # campaigns for both activities, supervisors and telcon; # minutes/call for reps

USING DRIVER-BASED DATA IN AN IES MODEL

Example: CAM -I's *The Closed Loop*

4. Separate fixed, step and variable costs and assign them to appropriate cost objects
 - Fixed = sundry (\$40k), lease (\$200k), general manager (\$100k)
 - Variable = reps, telcom
 - Step = supervisors (\$0 to 13.5 campaigns, \$120k (13.5 to 27))
5. Compute slope of variable portion of cost functions
 - Telcom = \$5k/campaign
 - Reps = activity consumption rate x resource consumption rate x cost/rep = 100k calls/campaign x (10 min/call x 1 hr/60 mins x rep/1500 hours) x \$50k/rep = \$555k/campaign
6. Apply capacity constraints
 - # campaigns for reps = 13.5

IES Model STRUCTURE: "Closed Loop"

