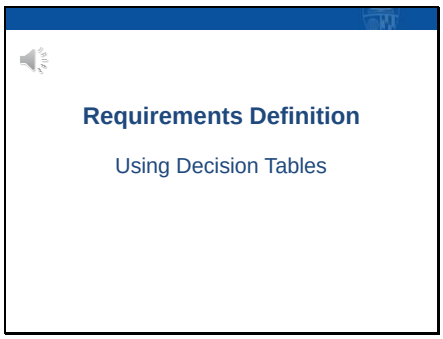
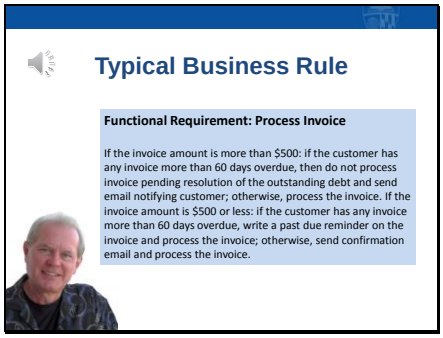
1



In this lecture we'll discuss how to use decision tables to document business rules.

2



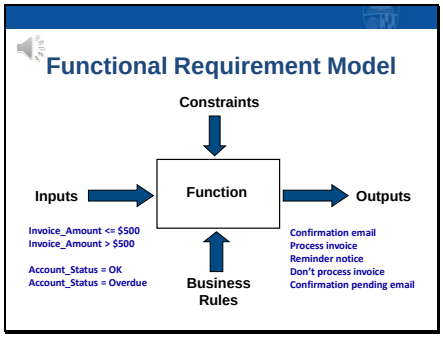
You may recall from an earlier lecture that many functional requirements have business rules associated with them. Business rules typically specify how to take a function's inputs and convert them to the function's output.

Here's a set of business rules for a functional requirement. The system requirement is to process an invoice. Depending upon the amount of the invoice and the customer's account status, the invoice will be processed and a confirmation email will be sent, the invoice will be processed with a payment reminder and a confirmation email will be sent, or the invoice will not be processed and an email will be sent to the customer explaining why.

These business rules are expressed in narrative form, which is the most common form used in practice. Expressing business rules using narrative is fine for simple rules, but the more complicated the rules get, the higher the risk that the rules will be documented incorrectly or interpreted incorrectly. A better technique for documenting complex business rules is to use a decision table.

So…let's see how that's done.

3



Let's start by recalling the functional requirement model introduced in an earlier lecture. Remember…a functional requirement can have up to five parts. When using a decision table to express a requirement's business rules, we typically start by identifying the requirement's inputs and outputs.

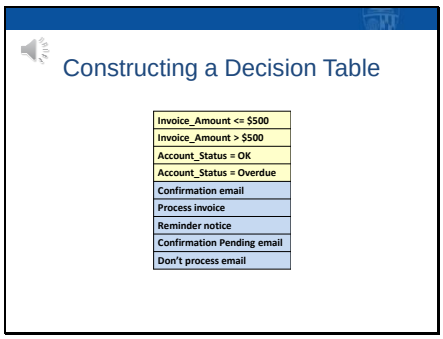
In our sample business rules there are two input variables…the amount of an invoice and a customer's account status. Different decisions take place depending upon whether the invoice amount is up to $500 or greater than $500… so we'll identify two conditions for that input variable as illustrated here. Similarly, there are two conditions for a customer's account status: the status is okay if there are no invoices outstanding for less than 60 days, and the status is overdue if there are any invoices outstanding for 60 days or more.

There are five distinct outputs: send confirmation email, process invoice, write reminder notice on invoice, don't process invoice, and send a confirmation pending email.

Depending upon various combinations of the inputs, different outputs will occur.

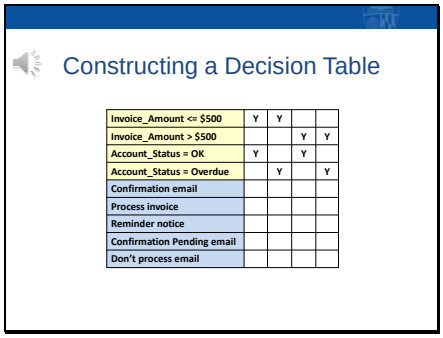
Now that we've identified the inputs and outputs, we're ready to construct a decision table.

4



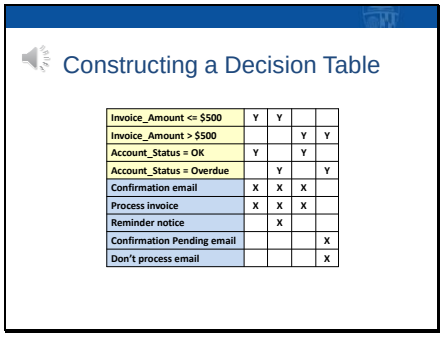
The first thing we'll do in constructing our decision table is to set up a row for each input condition and each output condition…like this. The inputs are shown in yellow and the outputs are shown in blue. So, our table has nine rows.

5



Next, we generate the different combinations of inputs that are possible. I've marked them with Ys. For example, the first column represents an invoice amount up to $500 and an account status of OK. The last column represents an invoice amount greater than $500 and an account status of overdue.

6



And then…for each set of inputs, we indicate which of the outputs will occur, by marking the appropriate cells with Xs…as illustrated here.

We read the decision table from top to bottom…with each column representing one possibility. For example, the second column indicates that if the amount of an invoice is up to $500 and the account status is overdue, a confirmation email will be sent, and the invoice will be processed with a reminder notice added to the invoice.

Now…the benefit of using a decision table is that it is precise. The likelihood of misinterpreting the business rules should be zero…much lower than when the rules are expressed using narrative.