ETA Week 3

The Session in Review

We began this session by asking for any questions that might have arisen when thinking about week 2 topics or while completing the assigned homework task, the task being to create a decision table from some information we had provided. The interesting part of this was that we hadn't covered how to create decision tables in week 2, we ran out of time and moved it to week 3. Despite that some of you made very good attempts at building a decision table.

From our opening discussion we spent a little bit of time going over decision tables and where they might be useful. We spoke about decision tables being a testing technique, a very useful one, but only when it is used in the right context. Lee and Paul both think, from the questions asked and suggestions provided, that you understand the reasons for decision tables and the concepts. We also suggest that you find ways to practice building them. Remember that the thinking is in identifying the conditions. Once you get those the rest is just filling in boxes.

After decision tables we spoke about "all pairs". This testing technique takes inputs that need to be tested and pairs them. We know that we often find bugs when pairs of inputs interact. When we pair test we take advantage of this. When we use a tool such as "all-pairs" we can also reduce the number of test cases we need to execute and not adversely impact risk. We demonstrated the software (a copy is in Github) and had a look at how, in my real world example, we could reduce test cases from 171 to 33 by pairing and removing duplication (the software does all this work for you). We also spoke about how this approach gives you options, not rules. You may decide to run more or fewer tests, you may decide to run other tests. You must evaluate the information and test as you think is fitting based on context.

The last thing we discussed in the session was Waterfall development. In coming sessions we will look at agile and compare and contrast.

Lee took us through the theoretical structure of waterfall. This includes ideas such as: software is built in "phases"

- the phases are (generally) business requirements, design, coding, testing, release
- assumed perfect knowledge of what the client really needs
- having to complete a phase before moving to the next phase
- teams are in "silos" (the Business Analysts, Developers, Testers all physically sit in different areas, they may even be in different companies)

The group asked some excellent questions and made some insightful observations during our discussion of waterfall. These included:

- it seems inflexible
- where are we including the clients?
- why is everyone sitting apart?
- it doesn't look like it is a good way of communicating
- do the clients really know what they want at the start?
- what if the client changes their mind part way through the project?
- are we talking to the clients enough?

We closed this discussion with a reminder from Lee and Paul this was not about saying waterfall is bad. It was about giving everyone an understanding of the waterfall fundamentals. Many places use this method to develop software so it is important you are aware of waterfall, why it is used and also, some of its problems. We both encouraged you to read a few web articles about waterfall to get some more information.

This was another great session. Thanks to everyone for taking part.