Testing

Part 1: When and What to Test

1. Early in the Development Process
2. Throughout the Development Process
3. The Final Check

Part 1 – Testing Early in the Development Process

- Overall Concept
- Program Survey(s)
- Graphic Layout and Design of Program
- Usability of Program (website, interactive tools)
- Images
- Messages

Part 2 – Testing Throughout the Development Process
3 Main Areas to Focus

1. Survey/Dictionary
   - Every question used for tailoring - in Dictionary as Characteristic?
   - Every response used for tailoring - in Dictionary as Value?
   - Characteristics and Values defined correctly in Dictionary?
   - Substitution text in Dictionary – appropriate? correctly spelled?
   - Derived Characteristics being calculated correctly?

2. Design Elements
   - Layout pages/sections/containers – correctly designed and sized?
   - Font selection – readable? right choice for content? too many?
   - Image – right choice to support text? all needed variations created?
   - Template names – match message sections! text/images appear?
   - Image names – do they match image?

3. Messages
   - 3 Areas to Focus on when Testing Messages:
     A. Length
     B. Readability
     C. Logic

A. Length
   - Needs to be kept in check for web programs
   - Has to keep in mind attention span for video/audio
   - Is limited by other media – i.e.: text message, email message
   - Is crucial to get right in print-based materials
     - Too long – text runs off page
     - Too short – text looks sparse, unprofessional

Length – What To Look For
   - Equal length for all Values tailored on from a Characteristic
   - Limiting number of messages for special cases
   - Standardize the number of messages people get
     - Can control using a List or Select command
     - Create default targeted messages when need to even out
Focus on ONE container at a time
- Look at the message set for that container
- Find your longest sentences and paragraphs
- Find your shortest sentences and paragraphs
- Create test cases that will result in the longest and shortest possible messages
- Tweak messages and re-test until they fit
- Test with "average" test cases to see impact of tweaks

Messages - Eye Balling It

Messages aimed toward main goal
- Content presentation follows a logical order
- Text is appropriate for given audience/topic (serious vs. funny)
- Literacy level is appropriate
- All selected messages flow together and make sense

Spelling errors
Grammar errors
Unclear sentences
Lack of page flow
Weak/No segues between sections where needed
Lack of introductions and closing statements

Reading Level of Messages
- SMOG Test and/or Fry Formula
- Spelling/Grammar
  - Editorial review of all messages
- Everything Else
  - Create 4-8 test cases
  - Make their sample data as varied as possible
  - Read output for all messages
  - Create many more varied cases for continued review
Readability – Where To Test

- Message Table
  - Monitor text as you write
- MTS Preview
  - Sample messages per test case – plain text
- MTS Publisher
  - Samples messages per test case – in your layout

Logic – What To Look For

- Typos in Logic – do you have error messages in Preview?
- Substitution text – is it there?
- Derived Characteristics – calculating properly?
- Message selection – is it the anticipated message?

Logic – How To Test

- Focus on creating cases that will allow every message/image to fire at least once.
- Run messages/Dictionary/test cases through Preview for error identification.
- Use MTS cases for future testing of database cases.

Logic – Where To Test

- Message Document
- Preview
- Publisher

C. Logic

1. Content appears in every section.
2. Content matches corresponding logic.
3. Priority systems, if present, are checked for accuracy.
4. All potential variations are covered.
5. Default messages are accounted for and appropriate.
When you feel the program is error free and working as anticipated, complete one final check of everything you’ve already tested for.

No fixes needed? Then give it 100% sign-off.

Keep on hand:
- 10-20 diverse test cases that can be run through MTS if any future changes/updates happen. If they continue to produce anticipated results after changes/updates – good!

What’s Next

- How to create test cases in MTS
- How to test messages in MTS using:
  - Dictionary, Message Table, Test Case
- How to review errors in Message Tables
- Fixing errors (hands-on project in the labs)