Development

Friday, 9 October 2015 10:30 AM

Development

Text pp296-301

Development has 4 parts:

Manipulation (coding)

- Create the solution by electronically manipulating (coding) the prepared designs
- Include internal documentation where necessary

Validation

Validation is ensuring inputted data is

- of the right type (e.g. numeric)
- reasonable limits (e.g. ages between 1 and 130!)

Databases, spreadsheets and applications can have validation rules built into data fields to reject unreasonable entries. Validation can include:

- **Existence**: is some essential data simply missing, such as a name?
- **Reasonableness**: does it seem logical? Checking hours are within reasonable limits (e.g. working over 40 hours a week may be questioned)
- **Consistency**: Checking for inconsistencies in surveys (e.g. a person claims to be unemployed but earns $80,000 a year!) Some surveys ask similar questions in different parts of the survey to check whether people are inconsistent with their answers
- **Data Type**: e.g. have letters been entered instead of digits?
- **Format**: e.g. an CustomerID must be three letters followed by 2 digits. A must date entered as dd/mm/yy?
- **Range**: is a date in August between 1 and 31?
- **Return to sender**: a user requesting to join an emailing group may be sent a confirmation email
- **Check digit:** eg: VCAA student number

Keep in mind that data can be valid but wrong!

And that validation is not testing; you **validate input** (for reasonableness) and **test output** (for accuracy)

Testing

All functionality of a solution using the test table created during Design is used to compare the expected output with the actual output. Testing can be:

- Informal testing - done as each new feature of a system is coded\added
- Module\component testing: done when a new module\component is developed it works as required by itself
- Integration testing: done to check that new modules\components interact properly with the existing modules\components
- Formal testing - done when the system is finished to test the system as a whole
- Reliability - does the solution:
    - Work as required: eg handle the flight of the drone in real time
    - Cope with the unexpected\unwanted: eg power spikes , brown-outs and back-outs
- Does the solution provide output that is:
    - Correctly formatted
    - Relevant
    - Accurate
    - Clear

Documentation

Useful documentation is:

- Clear: ie understandable in appropriate language
- Concise: short yet comprehensive with use of relevant images

- Complete: check key steps and warnings
- Current: ie timely
- Correct: no errors
- Easy accessible (which is the best type of documentation)
    - Quick start guide
    - Context sensitive help
    - Online help\tutorials\web
    - Printed manual
- Easily searched: users must be able to quickly find the information they need

oooOOOooo