

Building an Application Framework

Part I

Terry Bradbury
terry.bradbury@gmail.com
@Terry_Bradbury



pluralsight 
hardcore dev and IT training

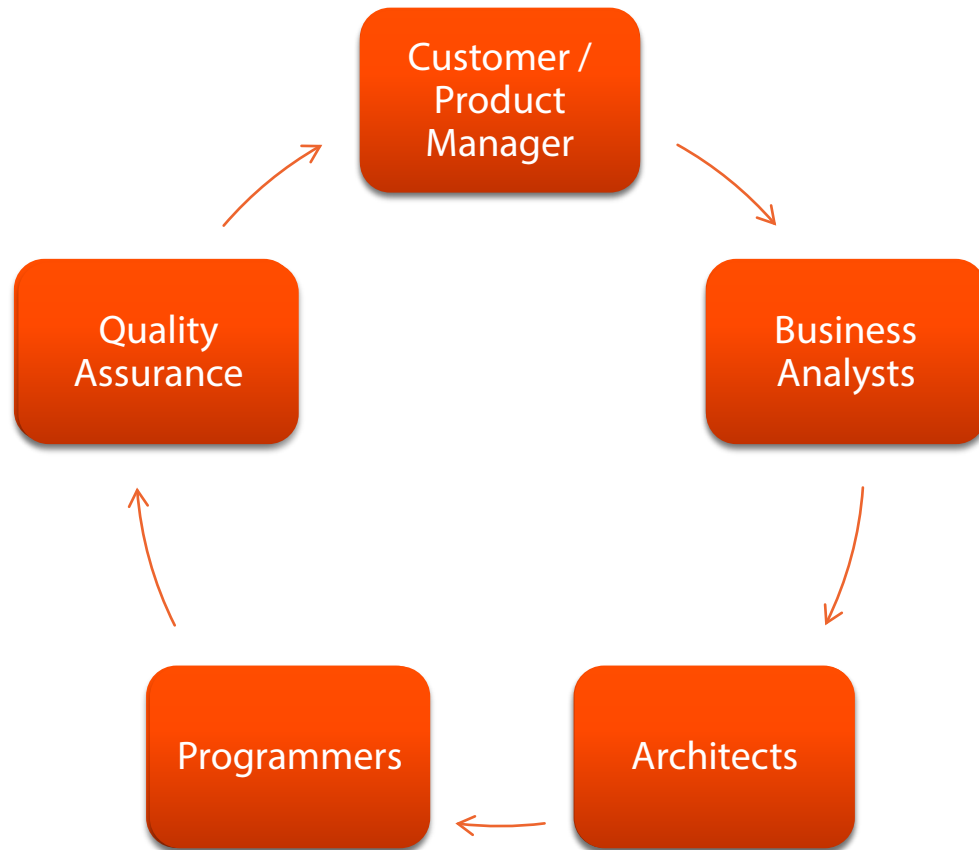
Overview

- The Development Cycle
- Multi-layered Architecture
- Library Management
- Function Libraries / Services
- Debugging / Logging / Error Handling

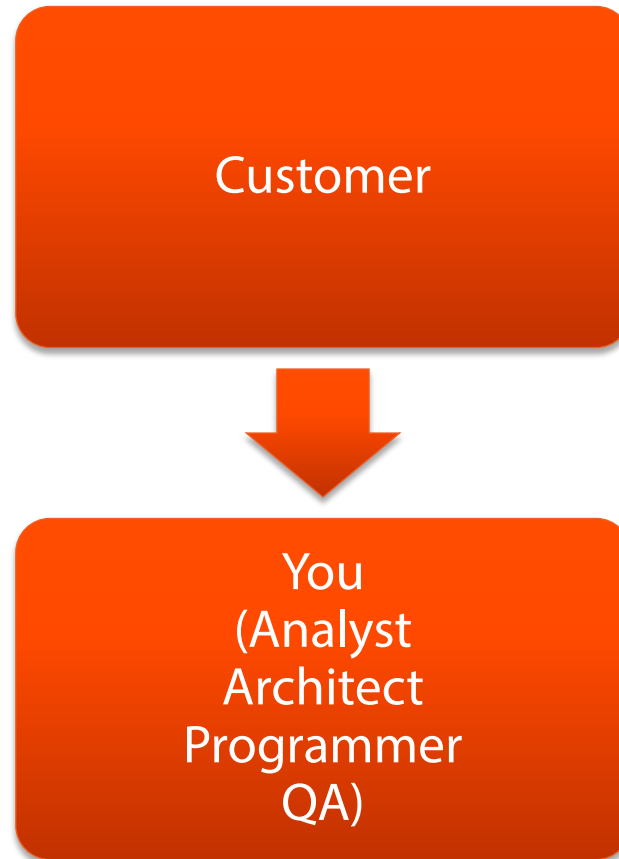


The Development Cycle

The Development Cycle



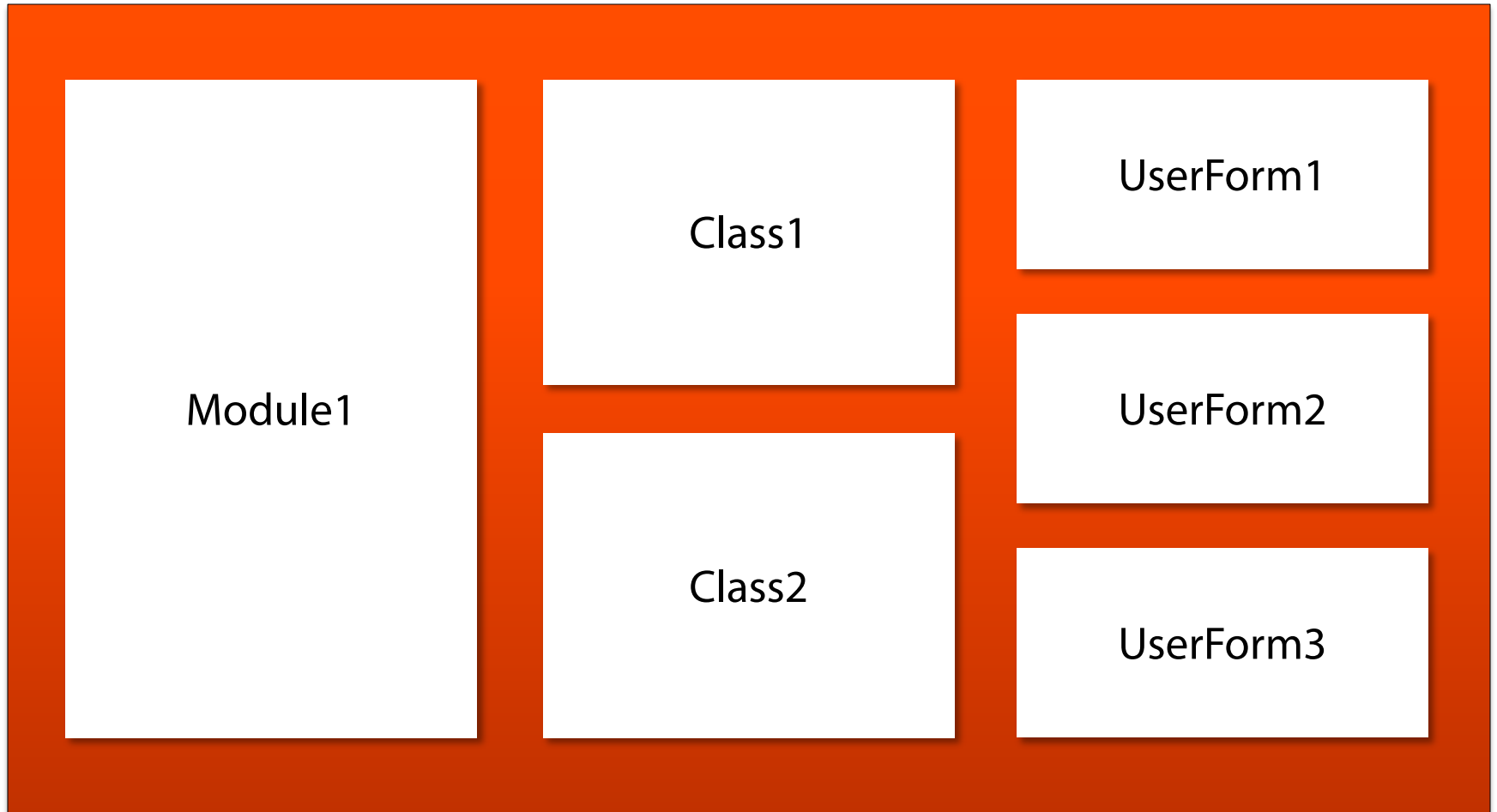
Office Development



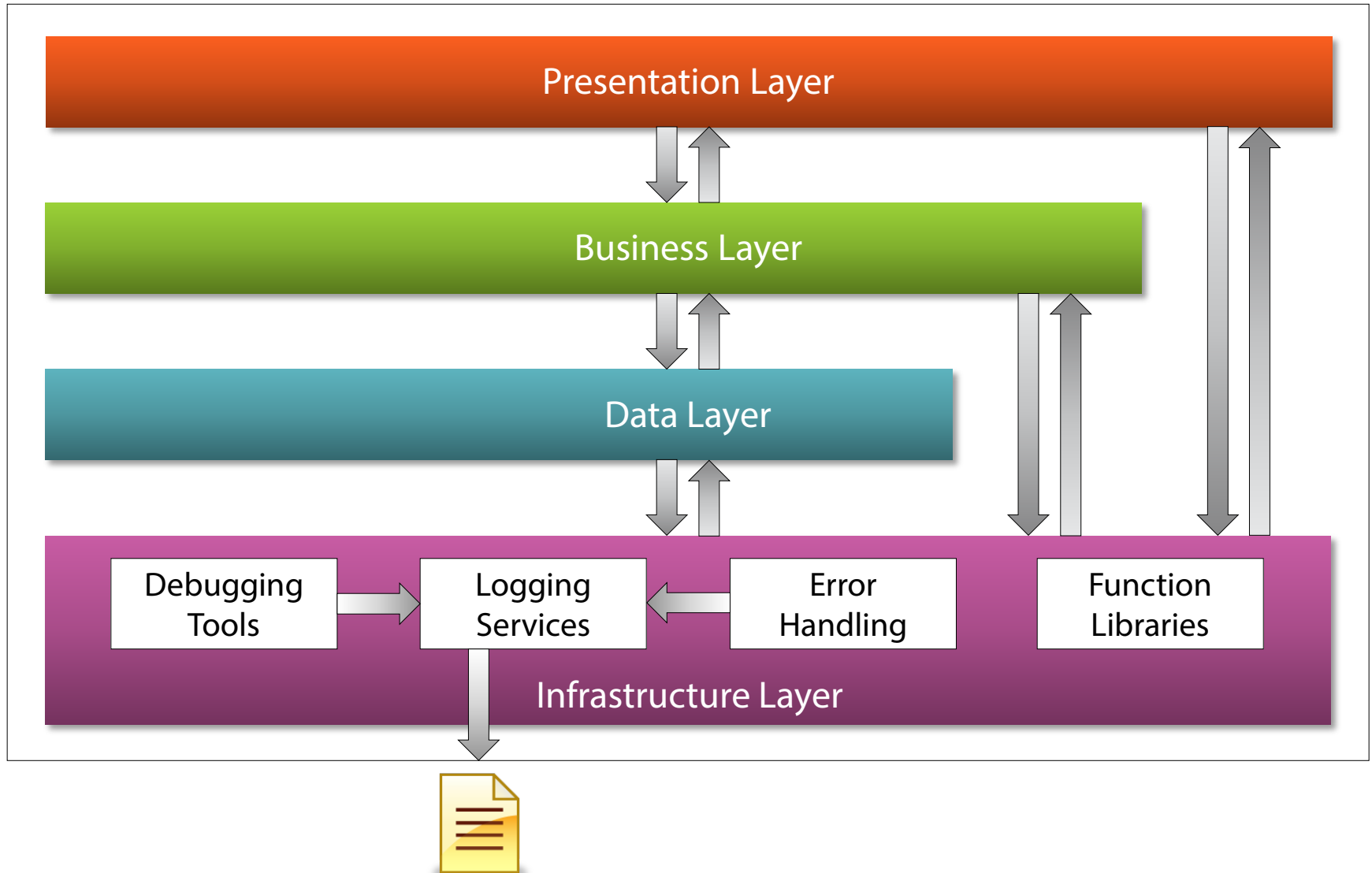


Multi-Layered Architecture

Monolithic Approach



Multi-Layered Architecture



Translating Architecture to Code

- **A system architecture is typically made up of layers**
- **Architectural Layers = groupings of libraries for implementation**
- **Use abstraction to reduce dependencies**
 - Dynamic references to worksheets
 - Use Interfaces to hide implementations
 - Opportunities for code reuse
- **Keep code within layer boundaries**
- **Think in terms of services, not procedures for reusability**



Library Management

Developing Standards

- Naming modules
- Naming procedures / members
- Documentation
- Code organization

Naming Modules

- **First by type –**

- M – Code Module
- F – UserForm
- C – Class Module
- I -- Interface

- **Then by app code or grouping**

- MVBA – general-purpose VBA libraries
- MMSOffice – Procedures for working with Microsoft Office apps
- MWinAPI – Windows API functions

- **Finally, specific purpose**

- MVBAStrings – Library of string functions
- MWinAPIRegistry – Functions for working with the Windows Registry APIs

Naming Procedures and Members

- **Generally two forms for 'action' procedures –**
 - VerbSubject (NewCustomer)
 - SubjectVerb (CustomerNew)
 - Works better for the Macros dialog
 - Hide non-entry-point procedures using Option Private Module
 - Casing Standards
 - NewCustomer or newCustomer
- **Retrieval functions often start with 'Get' (GetCustomerId)**
- **Boolean Properties (Capabilities or Attributes)**
 - Can, Has, Is prefixes
- **Custom Events to respond to property changes**
 - OnNameChanged

Documentation

- Document for the next developer
- A standard module header is the table of contents
- A standard procedural header identifies parameters and purpose
- Code can never be fully self-explanatory
- Comments should be focused on the why more than how or what

Code Organization

- **Placement of a procedure within a particular module or class**
- **Order of procedures or members within a module or class**
- **Code Modules**
 - Procedures in alphabetical order
- **Class Modules and UserForms**
 - Group by interfaces, properties, procedures, and events
 - Alphabetical within groups
- **Helpful when**
 - Comparing versions of a module
 - Looking for a member in source code outside of the editor
 - Moving code between modules
- **Use Sort Procedures utility in MZ-Tools add-in**



Function Libraries and Services

Standard VBA Modules

- **MVBAArray**
- **MVBABitwise**
- **MVBAConversion**
- **MVBADebug**
- **MVBAAError**
- **MVBAAFile**
- **MVBALogFile**
- **MVBALogic**
- **MVBAMath**
- **MVBAStrings**
- **MVBAUtils**

Bitwise Methods

Key #1 –

Write simple procedures that do one thing really well, then combine them to add needed complexity.

File Methods

Key #2 –

‘Wrapper’ functions can be used to ensure libraries are implemented in a consistent manner.

Logic Methods

Key #3 –

If a given type of evaluation is needed more than once, functionalize it for re-use.

String Methods

Key #4 –

Functions may be used in different contexts – don't limit yourself by solving just the task at hand.

Points to Consider

- **Libraries should not be all-encompassing**
 - Provide only what is needed
- **Don't "re-invent the wheel"**
 - Use the VBA developer community to help solve your coding problems
 - Learn from other developers' solutions



Debugging / Logging / Error Handling

Debugging

- Code tracing
- Code trapping
- Assertions
- Print

Tracing

Trapping

Logging

Error Handling

Summary

- The Development Cycle
- Multi-layered Architecture
- Library Management
- Function Libraries / Services
- Debugging / Logging / Error Handling