Workflows

Summary of Development Processes

Please find a summary of typical project lifecycle and my standard approach, workflows and processes towards development:

Typical Project Lifecycle

1. Initial Design Meetings: Account Manager, System Designer, myself and on occasion also the client.
2. GUI: Design and submit screenshots to client for feedback and sign off.
3. Program: Design, develop program and research system components as required.
4. Dependencies: Collate all dependencies including liaising directly with the client e.g. onsite network/IP.
5. Offsite Test – GUI/Program: Complete all testing of GUI/program using local RMC processor and X-Panels.
6. Offsite Test – Rack: Confirm control of all controlled equipment and AV signal integrity.
7. Onsite Commission: Fully commission system onsite.
8. Onsite Handover: Demonstration and handover to client.

Please find below a detailed breakdown of the above project lifecycle and the general approach taken in order to promote the following aspects:

1. Client Clarity: System fully meets client requirements and expectations with no miscommunication.
2. Changes: Minimise any potential last-minute changes to an already completed touchpanel/program.
3. Efficiency: Development is systematic, consistent and efficient providing the optimum product within the available time.
4. Commission: Commissioning onsite is smooth, efficient and completed within available time.
5. Fault-Free: System is entirely fault-free, operates as required and is signed off.

Initial Design Meetings (Step 1)

The purpose of the initial design meetings are as follows:

- Brief: Extract as much information as possible on how the system is intended to be used in order to build up brief.
- Options: Client may not be fully aware on how they may wish to use the system and may require appropriate guidance, options or suggestions.

- Agreement: Reach agreement with all parties on how the system should operate crucially in order to prevent the need for the client to request further changes at a later stage.

GUI/Program Development (Steps 2-5)

Touchpanel(s):

1. Evaluate system schematics.
2. Manually draw out on paper GUI pages, it is a lot quicker to change GUI layouts on paper rather than after they have been built.
3. Collate and feedback any queries, questions, concerns or potential issues with the current system design.
4. Build pages.
5. Send through to client for feedback if required.
6. Assign all Join Numbers to all pages ensuring:

    - Spacing between natural functional groups to allow for expansion if required in future.
    - Establish consistent natural numerical ordering and grouping of functional components within the GUI.

Program:

1. Construct system skeletal structure:

    - Declare/configure controlled external devices via RS-232, IR, IO, IP or by other means.
    - Declare/configure touchpanel(s) establishing consistent/logical numerical ordering of IP-ID's.

2. Join Numbers:

    - Define/populate all Join Numbers as defined on touchpanel(s) within program.
    - Define/establish naming conventions for all signal names.

3. Internals:

    - Build out all internal modules and logic.
    - Program must compile cleanly with no errors/warnings in order to help minimise potential bugs.

Dependencies:

1. Client:

    - Ensure that all client-side dependencies and considerations are collated and issued to client.
    - Ensure that the client is provided with as much time as possible to collate, respond and provide any such dependencies e.g. network/IP, room names or any other system specific information.

2. DSP:

    - DSP configuration may be produced by a separate audio commissioning engineer.
    - Collate all preset/macro/microphone names and behaviours, ensure full clarity and coordination as to the required operation of any audio DSP's.

3. Local Network:

    - If the system contains a local IP network which does not reside on the client's network, then standard/consistent IP addressing should be employed:

192.168.1.10	Processor
192.168.1.20	Touchpanel
192.168.1.21	Touchpanel
…	
192.168.1.30	Matrix
192.168.1.40	DSP

    - The above IP address convention generally allows for clean and clear organisation of functional groups while also allowing for expansion at later date.

Logic/Graphical Test:

1. Local System:

    - Setup a local networked system on desk using a processor e.g. RMC3/4.
    - Touchpanel compiled *.vtz contains a built in X-Panel to enable full logic/graphical test of program.

2. Logic/Graphic Test:

    - Upload compiled program into local RMC3/4.
    - Use X-Panel in conjunction with the Toolbox/Debugger to fully test all logic/graphical aspects.
    - Use placeholder text for any unknown commands e.g. 'Samsung LCD Power On' to monitor commands are being sent.
    - All logic/graphical aspects must be fully complete and tested e.g. page flips, button/text/slider/progress feedback.
    - Time is always short onsite therefore:

        - No logic/graphical aspects should still need to be completed when onsite.
        - All logic/graphical testing should be completed before the onsite commission.
        - The only aspects that should need to be completed onsite are:

            - Control of kit: Confirm wiring and control of all kit to be controlled.
            - Timings: Derive any system timings e.g. system power on/off progress bars.
            - Commission: Overall test and final commission of all permutations of use.

        - This is to ensure that works are cleanly defined, understood and completed within the allocated timeframe.

Rack Test (Steps 6)

The purpose of the rack test is as follows:

1. Control of kit: Confirm wiring and control of all kit to be controlled.
2. AV Signal Integrity: Confirm the AV signal integrity of all kit e.g. video and audio are all as required.
3. Research/Evaluation: Confirm the required control, operation of any new, unique or unfamiliar kit.

Onsite Commission and Handover (Steps 7-8)

The time onsite is always short and prone to unforeseen events e.g. room unavailability, unconfigured client supplied codecs, IP/network issues or delay in any other client supplied dependencies.

Therefore it should always be aimed to keep the amount of work that needs to be done onsite to an absolute minimum, ensuring that as much work is completed offsite as possible.

1. Onsite Commission: The only aspects that should need to be completed onsite from a programmer perspective are:

    - Control of kit: Confirm wiring and control of all kit to be controlled.
    - Timings: Derive any system timings e.g. system power on/off progress bars.
    - Commission: Overall test and final commission of all permutations of use.

2. Onsite Handover: Fully demonstrate the system to the client/end user.

File Naming and Archiving

Please find my general approach for the naming and the archiving of source code and touchpanel files:

- The main purpose of the naming convention of any archived files is the ability to identify its corresponding system many years into the future without having to manually inspect what is running on the system.

- The only aspect which is generally immutable over large periods of time and can easily be described by the client over the phone is the physical location of the system, therefore the naming convention should ideally be as follows:

    Client – Location – Floor – Room Name – Version

    BBC - London - New Broadcasting House - Floor 9 - Room 9.45 - v1.2.zip
    BBC - Salford - Floor 10 – Maxwell Room - v1.zip
    BBC - Manchester - Brindley Building - Reception - v1.zip

    The above is clean and clear providing high confidence as to the corresponding physical system.

- Avoid:

  'AV4254-BBC-BrdRoom-v1.0001'

  - Includes installation Job Numbers which are meaningless many years into the future.
  - Includes unclear/vague description of the location and corresponding system.
  - Includes a generic 'BrdRoom' term which may or may not actually correspond to a boardroom.
  - Includes unconventional versioning.

- There should be only x1 archive file per system which should contain all compiled and uncompiled files of the system so that should a component fail the exact compiled files can be restored into the system.

- There should not be a folder containing a large array of random files with very little confidence in whether they are up to date and whether they correspond to what is actually on the system.

System Rewrites

On occasion existing systems can be extremely buggy where it can be better to perform a rewrite, in which case the touchpanel layout remains unaltered but the underlying processor program is rewritten. The process broadly involves the following:

Onsite Assessment:

- System is fully assessed.
- Manually inspect and identify which components are working and those which may need replacing.
- Manually draw up using pen and paper the current system design.
- Manually switch sources to identify the current system design if necessary.
- Manually inspect and document the required functionality of the overall system.
- Manually detect serial commands from COM ports if necessary.

Offsite Rewrite:

1. Touchpanel: Wipe all existing Join Numbers keeping only all existing graphical elements.
2. Program: Rebuild program as required.

3. Test: Upload into local RMC3/4 processor via a local network and use X-Panel in conjunction with Toolbox/Debugger to complete logic/graphical test.

Onsite Upload and Commission:

1. Backup: Fully backup the existing system.
2. Processor: Upload the new program.
3. X-Panel: Use X-Panels of new touchpanel files to test before permanently uploading to the touchpanels.
4. Touchpanels: Once new program and X-Panels are fully tested and the system is operating as required permanently upload to the touchpanels.
5. Handover: Demonstrate and handover newly rewritten system.

Remote Assists

Remote assists were a highly convenient and flexible means of assessing existing distant/remote systems and if required sending through an program or touchpanel update:

1. System Issues: An existing system in Scotland has reported issues.
2. Remote Update: Local engineer who covers Scotland connects to system directly.
3. Remote Login: Using the engineer's phone or other internet connection I remote into engineer's laptop.
4. Inspection: Coordinating with engineer we inspect and diagnose issue.
5. Update: If necessary I amend and send through an update providing a quick means of resolving issue.