RivelloMultimediaConsulting.com
info@RivelloMultimediaConsulting.com
001.310.306.1436

Proposal for

Application Architecture

ARCHITECTURE PLAN PROPOSAL

The purpose of this proposal is to describe the scope and context of the Rivello Multimedia Consulting (RMC) Architecture Plan (ArcPlan) to be offered for this project.

The proposed architecture deliverable fits within the 3rd step of the 7 step Software Development Lifecycle (SDLC). An overview of the steps is shown below. The ArcPlan will include 'System Design' deliverables for documentation (3.1), standards (3.2), and architectural deployment (3.3).

Notes:

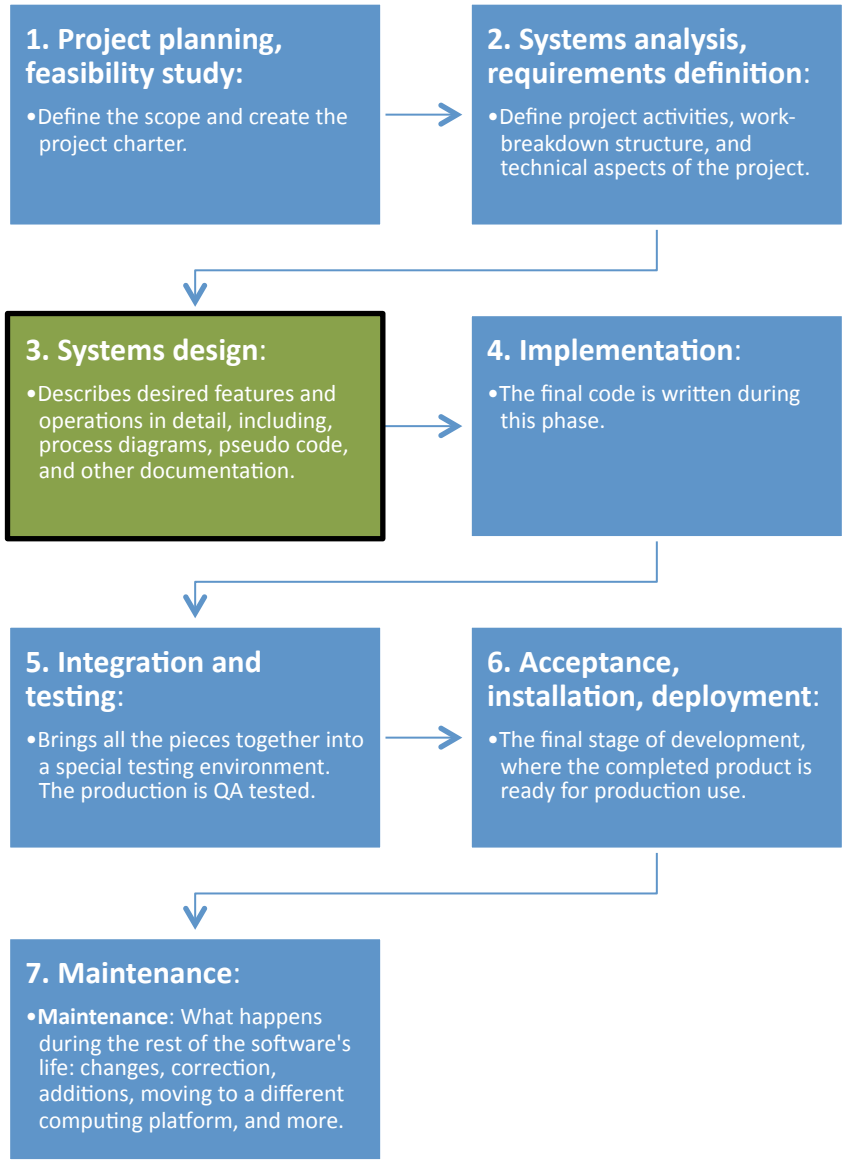
- **Training** from RMC is available on many related subjects. Some areas have been marked ("*") throughout the descriptions below. Training is not included by default in any estimates (time, cost, etc...) associated with the ArcPlan.
- A few **requirements** are mentioned in stage 1 & 2. These deliverables from the client for RMC inform the ArcPlan's creation. Typically the ArcPlan does not begin until receipt and review of these items.
- A milestone of note is stage 3.1.3's **Draft of ArcPlan Scope**. This is a great opportunity to review progress so far and discuss the breadth and depth of the final deliverable.

SOFTWARE DEVELOPMENT LIVECYCLE

Description:

The proposed architecture deliverable fits within the 3rd step of the standard Software Development Lifecycle (SDLC). An overview of the steps is shown here for context.

Steps:



1. PROJECT PLANNING & 2. REQUIREMENTS DEFINITION

Description:

These first 2 steps are outside the scope of the architecture deliverable, but are included for context.

Requirements: The client will provide a few deliverables from these steps to RMC before the ArcPlan assignment can begin. The format for these may vary from the description here, depending on client practices.

1. Complete Wireframes
    - **Wireframes** (http://en.wikipedia.org/wiki/Website_wireframe) are a basic visual guide used to suggest the layout and placement of fundamental design elements in the interface design. Because of this they must be completed before any artwork is developed. When completed correctly they will provide a visual reference upon which to structure each page. Wireframes also allow for the development of variations of a layout to maintain design consistency throughout the site. This is an important part of the initial development stage because it creates user expectations and helps to develop an awareness and familiarity throughout the site.
2. Complete User Stories
    - A **user story** (http://en.wikipedia.org/wiki/User_story) is a software system requirement formulated as one or two sentences in the everyday language of the user. User stories are used under the *Extreme Programming* (XP) paradigm for the specification of requirements (together with acceptance tests) (XP). Each *user story* is limited, so it could fit on a small paper note card — to ensure that it does not grow too large. The *user stories* should be written by the customers for a software project and are their main instrument to influence the development of the software.

3. SYSTEM DESIGN

Once the project has been defined ('the what' of the functional application) the ArcPlan assignment will answer 'the how', focusing on client-side architecture. The substeps are outlined below.

3.1. ARCPLAN CREATION

Description:

The ArcPlan is the core deliverable for the architectural consulting assignment. The ArcPlan includes the implicit recommendation for the PureMVC (http://puremvc.org/) architecture. **PureMVC** is a lightweight framework for creating applications based upon the classic Model, View and Controller concept. In a rare case, if PMVC is deemed an inappropriate choice, an alternative will be discussed prior to beginning the assignment.

The focus of the ArcPlan is on the client-side, but it will also propose how the client interfaces with the server (i.e. timing of server calls within the application's flow and an outline of the content for those calls).

There are three Unified Modeling Language diagram (UML) deliverables, described in 'Sections' below. 1) The *class* diagram of the PMVC 'actors' showcases many, but not all, of the major public members. This leaves the remaining public members and all private members and non-actor classes to be defined during development. The name and type will be given for each member listed. The Controller design-philosophy will be one that maximizes modularity of functionality. 2) The User *Activity* Diagram (UAD) references just how the actors in the UML handle application flow. Broadly speaking it resembles a flowchart. 3) The Client-Server *Interaction* Diagram (CSID) provides the high-level interplay between client and server.

RMC Phases:

1. Receipt and digestion of all available project specification documentation

2. Refinement of architectural strategy (iterative Q&A with stakeholders on front-end and back-end philosophies and conventions)
3. **Drafting of the ArcPlan Scope** and revision as needed to meet the approval of stakeholders. Depending on the project, this draft typically represents 10 - 30% of the information contained in final deliverable. The draft is intended to:

    - Describe the top-line architecture through drafts of the diagrams.
    - Help planning of the client-server setup and server setup, however ArcPlan mainly focuses on the client-side.
    - Inform the breadth and depth of the final ArcPlan deliverable.
    - Inform the client's timeline and resources for the development stage
    - Introduce basic PMVC concepts. This overview is ideal for stakeholders such as tech leads and project managers.

4. Completion and delivery of the ArcPlan
5. Revision of ArcPlan to meet stakeholders' approval.

Sections:

1. PMVC Actors *Class* Diagram (ACD)
    - MODEL
        - Objects
            - Properties
            - (Change) Events
        - Proxies
            - Properties
            - Methods
            - Notifications sent
            - Notifications observed
    - VIEW
        - UI
            - Properties
            - Methods
            - Events
        - Mediators
            - Properties
            - Methods
            - Notifications sent

        - Notifications observed
    - CONTROLLER
        - Application Façade
            - Notification-Command mappings
        - Commands
            - Where function is unclear, a brief description and a list of the major actors involved will be provided.
2. User *Activity* Diagram (UAD)

    - Opening 'overture' spanning from application initialization to 'awaiting-user-interactivity' state
    - Major user interactions stories (a diagram for each)

3. Client-Server *Interaction* Diagram (CSID)

    - Opening 'overture' spanning from application initialization to 'awaiting-user-interactivity' state
    - Major user interactions stories (a diagram for each)

3.2. BEST PRACTICES DEFINED

Description:

Stakeholders choose which best practices are to be supported during the development of this application. These choices inform the architect's deployment deliverable (3.3), and set the tone for the lead developer throughout development. Once the plan is decided, exceptions are treated as just that - exceptions. Supporting all of these sections is generally recommended if stakeholders can provide the needed support (such as time).

Sections:

1. Supporting the ArcPlan Itself (We can discuss exceptions like where we see 'architectural improvisation' likely and what will the process be when it pops up.)
2. Version Control Used for Code Backup and continuous Integration *
3. Code Standards (The 'RMC ActionScript 3.0 & MXML Coding Standards' guide and file templates are available upon request)
4. Documentation-creation using AsDoc* (File templates are available upon request)

5. Test Driven Development (Using FlexUnit to create test suites)*. The TDD strategy will focus on a standalone test suites primarily for each major PMVC Proxy & Mediator.
6. Code Review to ensure standards compliance

*Training available

3.3. ARCHITECTURAL DEPLOYMENT

Description:

Here the ArcPlan documentation informs the completion of the production environment (a version-controlled, shared Flex Project).

Sections:

1. ArcPlan and PMVC architecture discussed with production team.*
2. ArcPlan realized with a new Flex 3 PMVC project setup within the production environment. This will publish a blank screen, warning-free and error-free, but will contain only a handful of application specific classes so far. This will be the environment on which the development stage begins.

*Training available

4. IMPLEMENTATION, 5. INTEGRATION, 6. ACCEPTANCE, & 7. MAINTENANCE

Description & Sections:

These final 4 steps are outside the scope of the architecture deliverable, but are included for context. Some highlights of step 4 where the application development occurs include;

1. CLIENT
    1. Convert Wireframes to Flex 3 UI components
    2. Convert the ArcPlan into the classes defined in stage I's UML, SLD, and SD diagrams.
    3. Implement test cases for application functionality
    4. **Implement the code to provide application functionality** (This encompasses the bulk of the development time, of course)
2. CLIENT-SERVER
    1. Reference the ArcPlan to implement client-server communications.
3. SERVER
    1. Reference the ArcPlan to server's data model and implementation.

SUMMARY

The requirements and deliverables for this project are summarized here. Here are the action steps listed by chronology and task owner.

A) COMPLETED

- (None)

B) TO BE COMPLETED

1. Client: Delivers all relevant project specifications, including assignment requirements [See Sections 1 & 2]
2. RMC: Signs Statement of Work (SOW)
3. RMC: Begins ArcPlan Assignment
4. RMC: Delivers ArcPlan Draft [3.1]
    - Format: *Enterprise Architect* project in *.ea and *.pdf format.
5. Client: Approval of ArcPlan Draft
    - Format: Conversation (if needed) & Summary Email
6. RMC: Delivers Completed ArcPlan [3.1]
    - Format: *Enterprise Architect* project in *.ea and *.pdf format.
7. Client: Approval of Completed ArcPlan
    - Format: Conversation (if needed) & Summary Email
8. Client: Defines Best Practices for this project [3.2]
    - Format: Conversation (if needed) & Summary Email
9. RMC: Delivers Architectural Deployment [3.3]
    - Format: *Adobe Flex 3* project