AMA Downtime Post Mortem 2017

Prepared by: Nylon Technology 4-19-2017

Contents

[Overview 1](#_Toc480389343)

[Current Website Hosting Overview 1](#_Toc480389344)

[Systems Development Life Cycle (SDLC) 2](#_Toc480389345)

[Development Phases 2](#_Toc480389346)

[SDLC Development Models – Typical 3](#_Toc480389347)

[Waterfall 3](#_Toc480389348)

[Agile (Current Development Approach) 3](#_Toc480389349)

[Project Development History 4](#_Toc480389350)

[Current Development Matrix 4](#_Toc480389351)

# Overview

This document serves as a reference document for the current website development processes. While it is important to understand the reasons in the recent site downtime episode, one of the key factors in the event was a lack of a clear understanding of the entire scope of the website, it’s hosting, configuration, project requirements, and other constraints usually involved in this type of work.

A single event in any system, particularly in software development, is often only a symptom of a larger process problem and so normalizing an understanding of this entire scope is necessary, including typical development methods, other terminology, project history, etc., for conversational reference.

*However, this document is a focused overview of the process, touching on major elements only. It does not include functional/technical specifications, definitive best practices for all parties, etc.*

# Current Website Hosting Overview

*See “****AMA Hosting Topology.docx****” for further details*

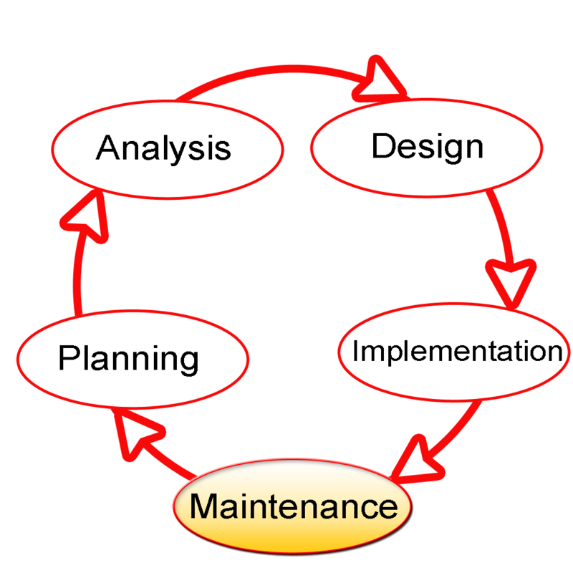
* 1. INetU Hosting
     1. Hosted at iNetU/ViaWest
     + General topology
       - Three Production web servers with a single clustered database
         * Two web servers load balanced
         * One web server as hot spare
         * Clustered database server shared by all three
       - One staging server
         * Combined application + database
     + RHEL 6.5 & 6.6
       - Red Hat Linux Enterprise, a common Linux based hosting environment
     + Apache 2.2
       - Open source web server running the majority of websites in the world today
     + PHP 5.3.3 PHP
       - Standard middleware language. The version is default for RHEL 6.5 & 6.6, but can be upgraded to higher versions
     + MySQL 5.5
       - Open source DBMS used by a vast majority of web sites, particularly in conjunction with MySQL
     + Drupal 7
       - Open source Content Management System (CMS) Framework deployed in the millions. Is customized with freely available pre-built, open source modules and can be extended with other custom modules. The Drupal core code and modules often require updates for features, security and performance

# Systems Development Life Cycle (SDLC)

The systems development life cycle (SDLC), also referred to as the software or application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

## Development Phases

SDLC covers many topics and it is important to understand the *context* of the term as it relates to the current development work on [www.amalgamatedbank.com](http://www.amalgamatedbank.com)



There are generally three major facets of web development, each of them incorporating the general workflow above:

* 1. New Site Development

A brand new site with analysis, complete documentation (functional/technical specifications), wireframes, graphic design, etc.

* 1. Large Project Update

A substantial component update of an *existing* site. In terms of a Drupal website, this would often be a major Drupal core or contributed module update, or custom module development. It may also include a major change in the display elements of the website, such as a significant change in the layout and design of site (User Experience).

* 1. Maintenance Updates

The bulk of the work currently done on the site. These changes often reflect functionality associated with a single page or 2 to 3 pages. Normally, the Drupal core code and modules are not directly impacted.

## SDLC Development Models – Typical

### Waterfall

The waterfall model is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production push and maintenance.

* + 1. Pros:
       - Linear, well defined steps from inception through production, QA, and launch phases.
       - Often used where sever hardware and/or software constraints require this level of detail.
    2. Cons:
       - Does not suit rapid development life cycles.
       - Each step followed is time consuming, whether or not it is necessary. For example, a complete system diagnostic during any change may be necessary, regardless of the scope of the change.
       - Extended planning, design, and development dramatically increase costs
       - Does not allow for overlap of projects – each project is often considered a system wide project, required complete system testing

### Agile (Current Development Approach)

Agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods.

* + 1. Pros:
       - Rapid, modular development when time to market is critical
       - Flexible development cycles – starting, stopping, and restarting projects is done with relative ease.
       - Loosely based on Waterfall as planning and procedure is essential to success
       - Linear, well defined steps from inception through production, QA, and launch phases, although overall scope is traditionally less than Waterfall.
       - A web development standard where rapid cycles of development and change are necessary
       - Documentation often meets absolute requirements only, saving costs for current refactoring of existing documentation
    2. Cons:
       - Can take multiple inputs that require extensive coordination
       - Typically, an entire system is not tested prior to production – only the aspect of the system being updated
         1. Note: the majority of changes in web development are not architectural in nature – often they are more cosmetic.
       - Short inception, design, and production timelines increase risk
       - Documentation is minimal and relies on already embedded knowledge

# Project Development History

1. **Spartan Development**
   * 1. 2014(?) > March 2015
     2. Developed Drupal platform as brand new website
     3. Planned and implemented InetU hosting topology
     4. Minimal documentation and knowledge transfer available
2. **Nylon Technology**
   * 1. March 2015 > Current (April 2017)
     2. Multiple Maintenance Projects performed
     3. Two Major Project Updates performed
     4. Server configuration upgrades planned (Apache + PHP)

# Current Development Review Matrix

This section outlines weak points in the overall process of development for the current website. Each of these elements will need to be refined in order to prevent additional issues.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Facet | Description/Problem | Related to  Downtime | Solution | Informs |
| Uptime Requirement | Understanding what percentage of time over a period of a year the site is expected to be usable. | Yes | Define. This is a critical step for all planning and requirements. | All aspects of SDLC, including hosting, planning, etc. |
| Production Push Time Periods | Understanding what times are available to push to production. | Yes | Define time periods and requirements | Project Planning, Resource Planning, etc. |
| Project Planning & Specification | Planning phases for projects often do not include major stakeholders from inception through beginning of production (AMA/Behavior/Nylon) | Yes | Focus on a phased approach to all projects, including all stakeholders | Resource planning, including launch dates/times. |
| Iterative Project Updates, Scope Creep | While the methodology is agile, there are often changes to the scope of projects during production, often removing any buffer before go live is necessary. | Yes | Although difficult because of approvals, last minute design changes impact delivery dates/times | Resource planning, including launch dates/times. |
| Project Priority Assignment/Collision | Multiple concurrent projects. Shifting dates often cause project development and go live dates to collide. | Yes | Develop fixed timelines, deliverables, QA, approval dates. | Resource planning, including launch dates/times. |
| Project Approval Process | Unclear project owner/stakeholder/team members | Yes | Document owner/stakeholder/team members | Resource planning, including launch dates/times. |
| Production Server Access | Current access via the remote desktop increases time and introduces connectivity risks because of dropouts. In addition, security credentials often require updating without warning. | No | Improve connectivity to production servers. Improve credential management | All aspects of SDLC, including hosting, planning, etc. |
| Change Management Process | A formalized process that identifies project pros/cons/risks. Usually involves documentation and signoff from stakeholders. | Yes | Develop documented approach to Change Management | All aspects of SDLC, including hosting, planning, etc. |
| Quality Assurance - Developer | QA/Test of new and existing code based on project and policy requirements. | Yes | Develop minimum testing constraints for all projects | All aspects of SDLC, including hosting, planning, etc. |
| Quality Assurance - Client | QA/Test of new and existing User Experience based on project and policy requirements. | Yes | Develop minimum testing constraints for all projects | All aspects of SDLC, including hosting, planning, etc. |
| Development Environment Requirements | Creating equal development, staging, and production environments to assure code functionality. | Yes | Base all development on Production specifications, where possible. Example: RHEL = CentOS as RHEL requires licensing, etc. | All aspects of SDLC, including hosting, planning, etc. |
| Resource Availability - Developer | Slipping project dates, changes in scope reduce resource availability. Go live dates/times also impact availability | Yes | Adhere to project planning dates. | All aspects of SDLC, including hosting, planning, etc. |
| Resource Availability - Client | Clients are often very busy and cannot be responsive to project scoping and QA tasks. This has pushed production and go live dates. | Yes | Develop method to resolve development tasks immediately. | All aspects of SDLC, including hosting, planning, etc. |
| Client Security Requirements | Security credentials are complicated. | No |  | All aspects of SDLC, including hosting, planning, etc. |
| Client Internal Communications | Project stakeholders/team members are occasionally out of synch with requirements or related projects. | No | Develop method to better inform stakeholders/team members | All aspects of SDLC, including hosting, planning, etc. |
| Blackout Dates/Times | Unknown blackout dates/times. This includes resource availability, holidays, critical site availability times, etc. | Yes | Develop method to record blackout dates/times | All aspects of SDLC, including hosting, planning, etc. |
| Testing Requirements | Unknown testing requirements and depth of testing requirements | Yes | Develop testing requirements; minimum testing, maximum testing, etc. | All aspects of SDLC, including hosting, planning, etc. |
| Project Ownership | Confusing project owner/stakeholders and team members | No | Document owner/stakeholder/team members | All aspects of SDLC, including hosting, planning, etc. |
| Website Fallback | Current hosting offers limited availability. | No | Develop fallback, other availability mechanism | All Scope |
| Platform Standardization | Development workstations, staging servers, and production servers must be on par with each other. | Yes | Ensure all touchpoints for development are equitable. | Testing/QA, development, staging, and production. |
| Team Communications | Although minor updates are relatively efficient, larger projects require complex interaction with third parties or AMA resources that have limited availability | Yes | Improve communications channels and require stakeholder/team member input | Testing/QA, development, staging, and production. |
| Testing Planning | While production may be complete, final Client QA can have large gaps | Yes | All stakeholders must be responsive to QA requirements. | Resource planning, including launch dates/times. |