Enterprise

Processes - Planning - Performance

Quad C™ Platform Technology Overview

Powerful Software Solutions:

Dated January 5, 2019 – version 1.2

ARCHITECTURE

SAAS (SOFTWARE-AS-A-SERVICE)

Quad C™ is a multi-tenant, multi-user, on-demand application platform providing true Software-as-a-Service (SaaS) delivery. No special or additional hardware is required to be fully functional. Clients are not required to maintain software or provide in-depth software/hardware administration. The application(s) can be securely accessed through any internet-connected device (Personal Computer, work station, or portable devices) equipped with a standard browser.

To provide scalability, ep^3, LLC ("ep^3") can add servers and instances without having to make changes to its architecture. Clients can configure the application with simple changes to their metadata. Finally, Quad C™ is multi-tenant-efficient, offering a load balanced farm of identical instances where each client's data resides on a separate database.

Real SaaS, Real Benefits

- Service not software means; no software to install or maintain
- Flexibility and ease of use
- No patches or upgrades – Quad C™ does not require patches, service interruptions, or manual upgrades. Clients enjoy real-time, instantaneous upgrades globally
- Enterprise implementation is measured in days and weeks not months
- Scalability - Companies of all sizes take advantage of real SaaS, scaling to hundreds of thousands of global users
- No infrastructure or installation costs
- No hardware associated maintenance expenses
- Work from anywhere anytime for global accessibility
- Upgrades or update would be delivered to all the customers as a result of the application's single configuration

Illustration1 – Basic platform design
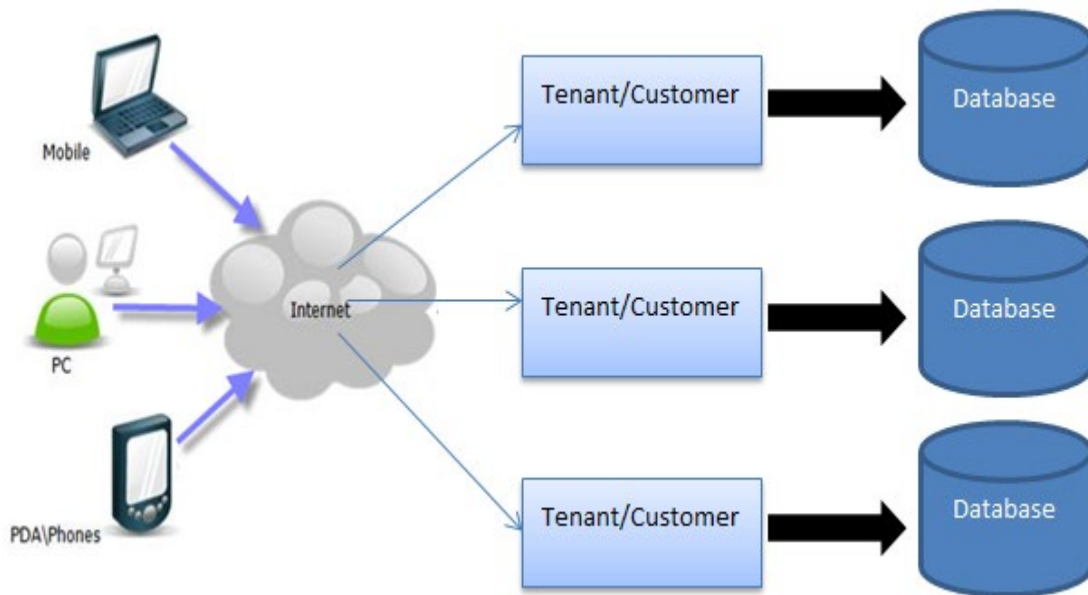
INFRASTRUCTURE

Quad C™ is a web-based n-tier application built on Microsoft server technology with an emphasis on Microsoft.NET and object development methodologies, which allows for rapid maintenance and expansion of features. The back-end business logic is built on an object oriented system, implementing a fully normalized relational DBMS design consisting of transact-SQL stored procedures on Microsoft SQL Azure.

TECHNOLOGY AND TOOLS

Quad C™ SaaS is built using Microsoft.Net framework and C# (C Sharp) language employing object oriented techniques. Quad C™ is modeled on a 3-tier architecture, which includes a User-Presentation, Business Logic and Data Access Layer.

All Data is persisted using Microsoft SQL Azure

Technology

- 100% Browser Based Application
- ASP.NET 4.0 and C#
- Microsoft Azure SQL Server 2012
- Asynchronous JavaScript and XML(Ajax)
- JQuery
- Web API (Rest Services)

Tools

- Microsoft Visual Studio 2010
- Microsoft Visual Source safe
- Microsoft AZURE SQL Server 2012

HOSTING

Quad C™ is hosted on Microsoft Cloud platform (AZURE). All hosted hardware and software patches are maintained by Microsoft. Patches required for Quad C™ is maintained by ep^3. Any core feature changes are for all tenants, which is an element of a true multi-tenant SaaS application. For unique client requirements and/or user location consideration; multiple instances can be created in different regions for better performance.

BACKUP

The protection of data is a primary concern to both ep^3 and its clients. ep^3 performs daily backups of the full database, stored on Azure cloud storage space specific for each client. The backup data is stored for 7 days and moved to external storage at an ep^3 facility. The backup data is accessible only when formally requested by a client. Application backup is done at ep^{3}'s development facility and stored at the ep^3 facility. Based on the client requirements hourly backup is performed for a day and stored at different cloud storage space than daily backup to avoid any data loss from single location.

DEVELOPMENT

ep^3 follows Agile methodologies defined in its Software Development Lifecycle (SDLC) to manage software enhancement releases and patches. The Agile methodology controls activities that take place within each release cycle including: enhancement requests; prioritization; project management; quality assurance; and, release management.

Our Software Development Life Cycle is composed of a number of clearly defined and distinct work phases used by systems engineers and developers. These phases are critical for planning; design, build, test, and delivery of the products. Our goal is to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems that move through each clearly defined phase, within scheduled time-frames and cost estimates.

The various stages of the lifecycle are briefly described below:

Requirement Analysis
Application requirements are established by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility studies in the economical, operational, and technical areas. After the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved by the management team and all stakeholders.

Design
Based on the requirements specified, usually more than one design approach for the product architecture is proposed and documented. This is reviewed by all of the key stakeholders and based on various parameters such as a risk assessment, product robustness, budget and time constraints, and design modularity. The best design approach is then selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Development or Coding

In this stage the actual product development begins and the product is built. The programming code is generated as per the design in the previous stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much difficulty. Developers have to follow the coding guidelines defined by the organization. The programming language is chosen with respect to the type of software being developed. The work is divided in modules/units and actual coding is started.

Testing

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. In this stage products defects are reported, tracked, repaired and retested, until the product reaches the quality standards defined in the Design Phase.

Deployment

Once the product is tested and ready to be deployed it is released to staging area where the testing is done again from test team and all stakeholders. Once the testing is approved the code is published to production environment for all tenants. If appropriate, product deployment happens in stages as per the business strategy.
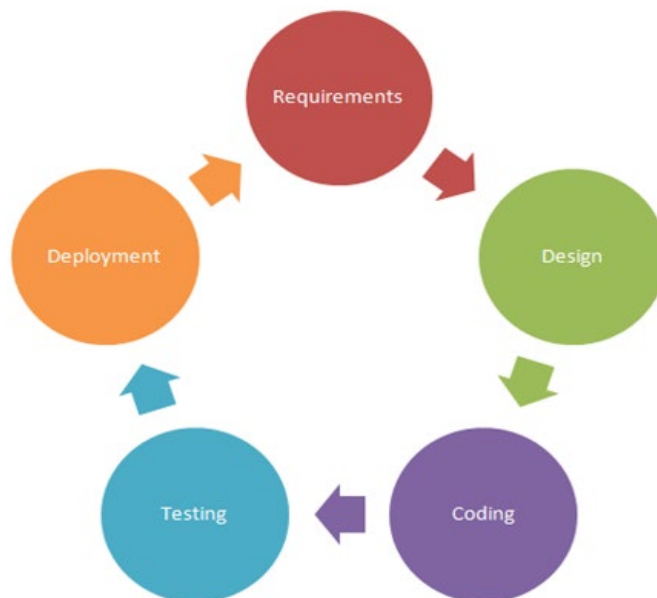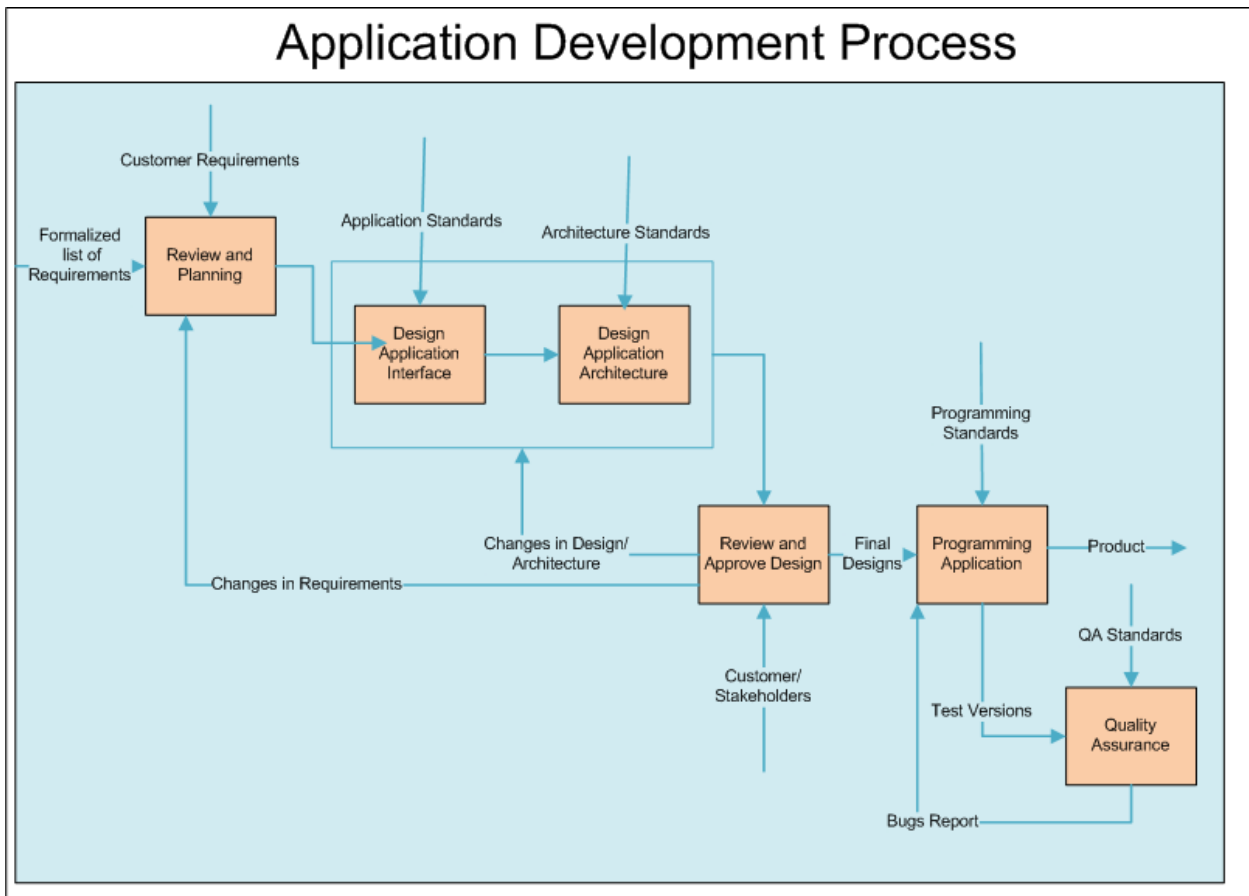
Illustration2 – Development Cycle

Illustration3 – Development Process

## Application Development Process

Development Standards

The goal of these documents are to create uniform coding standards among software personnel in the department so that reading, checking, and maintaining code written by different persons becomes easier. A mixed coding style is harder to maintain than a bad coding style. So it's important to apply a consistent coding style across a project. When maintaining code, it's better to conform to the style of the existing code rather than blindly follow this document or your own coding style.

Since a very large portion of project scope is after-delivery maintenance or enhancement, coding standards reduce the cost of a project by easing the learning or re-learning task when code needs to be addressed by people other than the author, or by the author after a long absence. Coding standards help ensure that the author need not be present for the maintenance and enhancement phase.

Quality Assurance Process

ep[3]'s software quality assurance process follows a methodology that focusses on end-to-end functional testing. The process involves a documentation phase followed by testing phase and release to staging and finally to production.

Illustration 4 – Quality Assurance Process
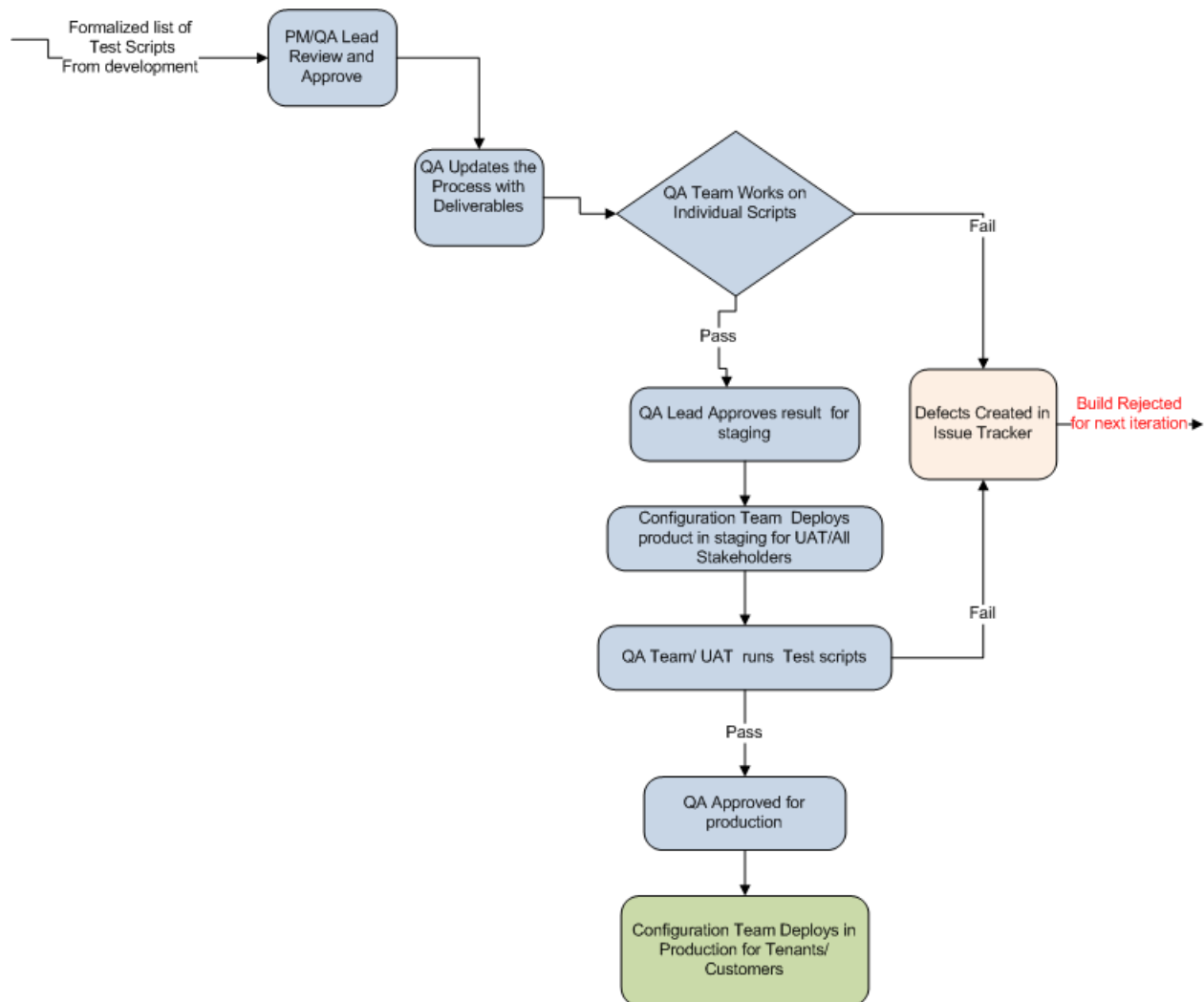
SECURITY

Access and Authentication

Security is important. Your data is secure with the same security standards in place for modern banking systems. Encrypted and unique logins by all users, data is stored and backed up on multiple servers located in different regions of the United States, and the application virtualized for each user with backup redundancy to minimize any potential downtime.

We ensure confidentiality, sender authentication, message integrity, by using IPSec (IP Security) for encryption and authentication of all IP packets at the network layer. We are committed to keeping our client's data reliable, secure and safe from malicious acts

All passwords are stored in the database in an encrypted format to minimize security liability. Passwords are hashed so that even ep[3] support or database administrator teams cannot lookup in database

Clients may get view only access to certain views in the database so that they can export the data and build custom reports.

All Data access credentials are maintained and encrypted in database and changed every quarter or when needed to avoid any vulnerability and malicious acts.

Application Permissions and Roles

Quad C™ platform is entirely based on roles. All the application data can be viewed by any authenticated user but only certain users who have rights based on the roles can edit or add data, approve tasks etc. Security roles for users are managed by Client Administrators. All roles for each application are predefined based on specific scenario on each module. ep[3] designates one system administrator at initial setup who can manage the system and security roles for their users. Additional system administrators are assigned by ep[3] based on clients request once approved.

<u>Internal Access to Production Data</u>

A limited number of ep³ Information Technology personnel, including Database Administrators, System Administrators and production support members are provide with access to the production SQL Databases as needed. This access is requested by customers or identified through ep³ monitoring procedures. Production access activities include the following:

- Production troubleshooting
- Researching data flow for client data
- Bug fixes
- Implementation of patches
- Client data change requests

All IT personnel access to customer databases is logged and audited on a periodic base.

Any questions concerning the *Quad C™ Platform Technology Overview* should be forwarded to info@ep3llc.com.