TECHNICAL PROPOSAL DOCUMENT  
FOR

Revision: 1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DOCUMENT CHANGE LOG | | | | | |
| Revision | Description | Author | Date | Approved | Date |
| 1.0 | Initial | Phuc Ngo | Jul 6, 2011 | An Cao | Jul 6, 2011 |
| 1.1 | Update the 2.1.1.3 Cleanse Product Info session to describe how the Product Info Cleanser processes raw data and suggested approaches.  Update some typo mistakes | Phuc Ngo | Jul 11, 2011 | An Cao | Jul 11, 2011 |
|  |  |  |  |  |  |
| This document has been generated from template TL-SSW-041 Revision 2.0 | | | | | |

Table of Contents

[TECHNICAL PROPOSAL DOCUMENT FOR WEB DATA EXTRACTION SYSTEM 1](#_Toc298165253)

[1. Project Definition 4](#_Toc298165254)

[2. Scope 4](#_Toc298165255)

[2.1 Project Requirements 4](#_Toc298165256)

[2.1.1 Functional Requirements 4](#_Toc298165257)

[2.1.2 Non-functional Requirements 6](#_Toc298165258)

[2.2 Deliverables 7](#_Toc298165259)

[2.2.1 Software Requirement Specifications Document 7](#_Toc298165260)

[2.2.2 Source Code / Database Schema 7](#_Toc298165261)

[3. Technical Specifications 7](#_Toc298165262)

[3.1 Overall Architecture 7](#_Toc298165263)

[3.2 Development Tools 9](#_Toc298165264)

[4. Team Organization 10](#_Toc298165265)

[4.1 Team Structure 10](#_Toc298165266)

[4.2 Team Definition and Roles 10](#_Toc298165267)

[4.2.1 Project Manager 10](#_Toc298165268)

[4.2.2 Technical Supervisor/Team Leader 10](#_Toc298165269)

[4.2.3 Quality Assurance Engineer 11](#_Toc298165270)

[4.2.4 Release Management 11](#_Toc298165271)

[4.2.5 Developer 11](#_Toc298165272)

[4.2.6 Quality Control Engineer 11](#_Toc298165273)

[5. Project Acceptance 11](#_Toc298165274)

[6. Meetings, Briefings and Reports 12](#_Toc298165275)

[6.1.1 Meetings and Briefings 12](#_Toc298165276)

[6.1.2 Progress Report 12](#_Toc298165277)

[7. Software Development Process 12](#_Toc298165278)

[8. Effort and Schedule 13](#_Toc298165279)

[9. Assumptions 14](#_Toc298165280)

# Project Definition

Based upon requirements in the Web\_Data\_Extraction.docx document, we understand that you are looking for a Software Development Service to develop a system named Web Data Extraction System (hereafter it is called WDE System) to develop a system to identify unstructured data sources on the Web, then store, analyze, and transform the sources into structured data to generate trend analysis reports.

# Scope

## Project Requirements

### Functional Requirements

Generally, the requirements of extracting and transforming product data from the sites and then providing data to Tableau tool are visualized in the following diagram:

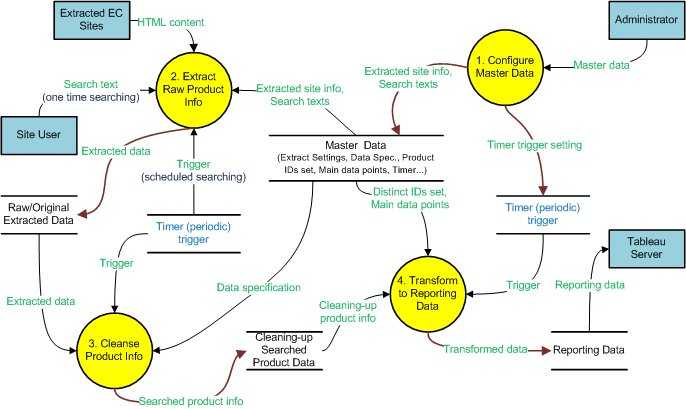


Figure 1 – Requirement Context Diagram

#### Configure Master Data

This is used to set up the prerequisite settings, configurations, used for extracting, transforming data and managing other information within the system.

The administrator needs to maintain the master information such as extracted site information, to enable/disable extracting an extracted site, to define the data specifications for cleansing raw data, … he will access the Administration Panel to do modifying on the system settings above. The Administration Panel is proposed some basic features as below:

* Extraction Source Maintenance
* Product Identity Set Maintenance
* Cleansing Data Specification Maintenance
* Other Settings Configuration (will be detailed later)

Moreover, in order to ensure the security for the Administration Panel, the system must securely identify system’s users and control the access to secured resources with authentication and authorization as well as data validation/input filtering.

#### Extract Raw Product Info

The Raw Product Info Extraction of WDE System can run unattended once a searching keyword has been entered by the Site User or periodically run as scheduled task. It will function to extract thousands of product data from 15 - 20 predefined eCommerce sites, such as:

* Amazon.com
* Walgreens.com
* Drugstore.com
* Cvs.com
* PlanetRX.com
* …

The product information will be split via pattern (regular expression) matching the keyword and then it is stored in a database structure like this:

1. Extraction Table
   1. Id int IDENTITY(1,1) NOT NULL,
   2. ExtractionSourceId int NOT NULL,
   3. SearchText varchar(max) NULL,
   4. CreateDate datetime NOT NULL,
2. Extraction Results Table
   1. Id int IDENTITY(1,1) NOT NULL,
   2. ExtractionId int NOT NULL,
   3. Rank int NOT NULL,
   4. Title varchar(max) NOT NULL,
   5. Price varchar(50) NULL,
   6. Url varchar(max) NULL,
   7. Thumbnail varchar(max) NULL,
3. Extraction Source Table
   1. Id int IDENTITY(1,1) NOT NULL,
   2. Site varchar(50) NOT NULL,

#### Cleanse Product Info

The Product Info Cleanser is a process used to determine inaccurate, incomplete, or unreasonable data and then improving the quality through correction of detected errors and omissions. The process may include format checks, completeness checks, reasonableness checks, limit checks, review of the data to identify outliers or other errors. Validation checks may also involve checking for compliance against applicable standards, rules, and conventions which are predefined in Data Specification dataset.

With the Administration Panel, users can configure the rules/conditions for the system to cleanse the searched product data based on the following approaches:

* Check data types of fields/columns with predefined specifications and their valid value ranges
* Standardize the data by converting text fields/columns to certain form (all camel-case or lowercase words), removing redundant blanks (multiple blanks to a single blank) and conjunction characters (such as blank-dash-blank to dash, blank-comma-blank to comma-blank, ...), removing most common abbreviations (such as copyright cycle, TM of trade mark) within the product name, converting prices in different currencies to a unique currency, rounding-off to common decimal fraction of numbers, recognizing and uniforming common units of weights and measures (pcs to pieces, m to meter, MByte to MB, ...).
* Check duplicates: simple matching like recognizing that Lenovo Y450 is a short name of Lenovo Ideapad Y450 (can base on matching conditions such as category attribute, first three letters and last three letters to narrow down the comparison list and then apply some fuzzy string matching technique) so that the system will group them as a unique product. The system will optionally support a confirming feature to allow users to confirm the matches (recognized by the system) and/or keep applying for the next runs (auto-apply without confirmation).

The Product Info Cleanser is triggered to process by a system timer. The trigger time is also configurable at Administration Panel.

#### Transform Data

Once extraction data is collected and cleansed, the Data Transformer will be functioned to periodically transform this data to some statistical summaries: mainly filtering/sorting the data by product across sites or by category, i.e. pet-food, within a site, or collecting the item-set patterns for market basket analysis, so on …

The transformed data is stored to a data warehouse that will provide data to Tableau Server to intentionally generate the trend charts/graphics for business intelligence purposes. It is triggered to run sequentially with Product Info Cleanser process by a system timer. The trigger time is configurable at Administration Panel.

There might be functionalities which are not mentioned on this document, will be analyzed and estimated when we have your detailed feedbacks.

### Non-functional Requirements

The WDE System is based on Windows multi-tier architecture guidelines. The following is prerequisite software for the operation of the system.

IMPORTANT: Internet Explorer v8.0 or higher is mandatory for the operation of the Web 2.0 homepages. The Web 2.0 homepages are not supported by IE 6 and unstable in IE7

#### Web Server

1. Windows 2008 R2, SP2 Enterprise Edition, 64-Bit
2. Internet Information Services (IIS) version 7
3. Microsoft .NET Framework version 4
4. Internet Explorer 9 WS2008
5. Microsoft URL Rewrite Module for IIS 7.0 (x86 version) (<http://www.iis.net/downloads/default.aspx?tabid=34&g=6&i=1691>)
6. WinSSHD 5 Installer (<http://www.bitvise.com/download-area>): to insure all access to the environments is over an encrypted connection

**Setting up an Email account:**

For sending email requirement, an email account should be set up on the client’s own email server. This account will be used to send mails from the WDE System application. WDE System does not act as a mail server, WDE System merely acts like an email client.

The external email details required to set up the WDE System email manager are:

* The mail server name/IP address
* Whether the mail server supports SMTP authentication
* Send port numbers
* Email address
* Email Account name
* Account User name and password

#### Database Server

1. Windows 2008 R2, SP2 Enterprise Edition, 64-Bit
2. Microsoft SQL Server Standard Edition 2008 R2
3. WinSSHD 5 Installer (<http://www.bitvise.com/download-area>): to insure all access to the environments is over an encrypted connection

## Deliverables

GCS shall submit following deliverables to Customer during lifecycle of the project, the time of the deliverables shall be defined exactly in project schedule when the project starts.

### Software Requirement Specifications Document

The Software Requirements Specifications (SRS) shall document GCS's understanding of the functional requirements as extracted from the SOW/Proposal and supplemental documents. This will be a streamlined version of a SRS due to the rapid development of the WDE System.

### Source Code / Database Schema

The final deliverable will be the completed source and database schemas and all supporting documentation.

# Technical Specifications

## Overall Architecture

WDE System is designed as a web-based multi-tiered application and it is segmented into a number of logical tiers: presentation services, business services, and data services. WDE System combines the latest client and server-side technologies to provide a robust, distributed application, offering ease of scalability and a higher level of security.

Examples of client technologies used include:

* ASP.NET Ajax toolkit (<http://www.asp.net/ajax/ajaxcontroltoolkit/samples/>)
* Web Part controls (<http://msdn.microsoft.com/en-us/library/e0s9t4ck.aspx>)

Cutting edge server-side technologies in use:

* Microsoft ASP.NET Framework 4 (<http://msdn.microsoft.com/en-us/netframework/default.aspx>) or Microsoft ASP.NET MVC (<http://www.asp.net/mvc>)
* ADO.NET Entity Framework   
  (<http://msdn.microsoft.com/en-us/library/aa697427%28v=vs.80%29.aspx>)

As new client and server technologies are available they are analyzed on an on-going basis to examine their potential to improve the user experience and architectural robustness of WDE System.

The WDE System application consists of 2 different tiers, as shown in the diagram below:

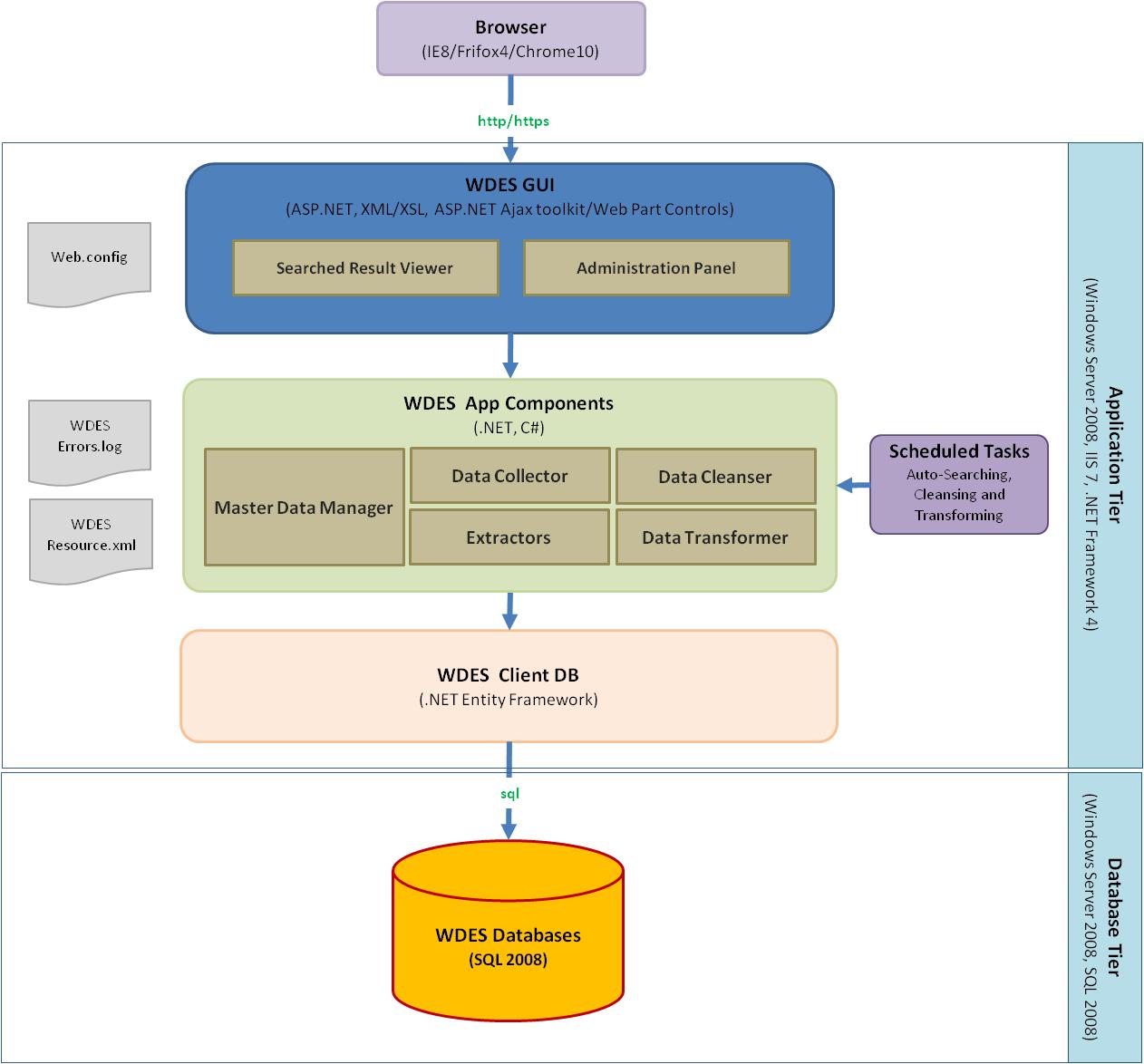


Figure 2 – Overall Architecture

Site Users (and Administrator) access the WDE System across a LAN, WAN or the Internet via a web browser. The browser communicates with the web server over any TCP/IP network using a lightweight protocol called HTTP that is the standard for the World Wide Web. The client sends HTTP Requests to the web server (IIS) and the web server responds by processing the request and sending an HTML-based page back to the user.

The TCP/IP network can range from a totally secure internal company network, a Virtual Private Network or a public network such as the Internet. Clients, who wish to use the Internet to connect to their WDE application but wish to maintain a secure connection, can use Secure Sockets Layer (SSL), which encrypts all data that passes between the client and the web server.

**Application Tier**

The WDES GUI components handle user interfaces, presentations on Searched Result Viewer and Administration Panel homepages. It might include some main elements or objects, as follows:

* ASP.NET Ajax Controls, ASPX (ASP.NET Active Server Pages) and Web Parts contain server-side scripting code that is executed on the Web Server before any HTML is returned to the client browser.
* XSL (Extensible Style sheet Language) pages are a transformation language enabling the display of XML (Extensible Mark-up Language) as HTML.

The WDES App Components encapsulate the business rules of the WDES application. All data extraction, transformation, validation, and updating logic for the WDES application are executed through CSharp.Net (C#) assemblies which run under the Microsoft .Net Framework v4. The Microsoft .Net Framework provides a large body of pre-coded solutions to common program requirements while managing the execution of programs written specifically for the Framework. It also supports standards based technologies such as Web Services enabling WDE System to be technology independent when integrating with other platforms.

* Master Data Manager: Handles functionalities related to controlling and maintaining master data
* Data Collector: executes proper Extractor to get product data matched the searched text, then stores extracted data to databases.
* Extractors: specified parse the HTML content that got from Extraction Source sites using regular expression that matches any inputted searched text.
* Data Cleanser: improves quality of extracted data by performing a number of checking rules and correcting or omitting inaccurate, incomplete, or unreasonable data.
* Data Transformer: transforms and aggregates clean-up product data to reporting data warehouse according to predefined transformation/aggregation rules.

The data access components (WDES Client DB) are responsible for communicating with the database and are able to provide independence from different Data Management Access Systems. They communicate with the database via OLEDB providers or the SQL Server .NET Data Provider as appropriate. Inside the components, the code relies upon Microsoft’s Entity Framework and/or ADO.NET data objects.

**Database Tier**

The Database Tier consists of data residing in 3 kinds of stores, 1 for the original data, 1 for data we are cleaning up and 1 for reporting. The Microsoft SQL Server 2008 relational database management system provides WDES data services. SQL Server 2008 provides a high-performance, highly-scalable engine for relational database data management and query processing.

The WDES databases might include a suite of Stored Procedures, Functions and Views that are called from the Application Tier data access components for all data manipulation.

## Development Tools

The following tools and software will be used to create the program during the period of WDE System project

|  |
| --- |
| **Server** |
| Windows 2008 R2, SP2 Enterprise Edition, 64-Bit  Internet Information Services (IIS) 7 |
| **Database Server/Client** |
| MS SQL Server 2008 Express Edition |
| **Development IDE** |
| Microsoft Visual Studio Professional 2010 |
| **Unit Testing** |
| NUnit Unit-testing Framework version 2.5.10 |
| **Code Quality** |
| FxCop for .NET Framework 4 built-in tool in Microsoft SDKs  StyleCop for .NET Framework version 4.3.3 |
| **Source Version Control** |
| Subversion (SVN) version 1.6.5 |
| **Continuous Integration** |
| Hudson Extensible Continuous Integration Server version 2.0.1  NAnt Build Tool for .NET Framework |

# Team Organization

## Team Structure

Project Manager

QA Engineer

RM Engineer

Technical Supervisor

Design Team

Designers

Team Leader

Development Team

Developers

Team Leader

Testing Team

QC Members

QC Leader

Figure 3 – Team Structure

## Team Definition and Roles

### Project Manager

The Project Manager will have both management and technical skills. His/her major responsibilities include:

* Coordinates interactions with the Customer
* Develop project plan, schedule
* Manage and control the project
* Report project progress to PM at Customer’s site
* Review work products developed by project team

### Technical Supervisor/Team Leader

The Team Leader will have mainly technical skills. His/her major responsibilities include:

* Be the Chief Architect of the system
* Design databases
* Design architecture – High Level and Low Level
* Monitor, control and support the developers
* Review all the security aspects of the system
* Support the Project Manager in technical decisions
* Review work products developed by project team

### Quality Assurance Engineer

The responsibilities of the QA Engineer include:

* Review/approve QA Plan
* Review project activities and progress against project plans
* Review project artifacts against standards/guidelines
* Audit project
* Identify and handle NCs during review/audit process
* Assist project team in following GCS processes
* Check project closure

### Release Management

The responsibilities of the RM Engineer include:

* C.I build support
* Develops Build Guides, Packaging Guides, Installation Scripts and Installation Guides with development team's support Generate software build(s)/release(s)
* Develop Build Note and Release Note if any

### Developer

The responsibilities of the Developer include:

* Develop application’s components
* Perform unit testing
* Fix bugs

### Quality Control Engineer

The responsibilities of the Quality Control Engineer include:

* Develop test-case document
* Regression test the system
* Verifying the bugs fixed/new features

# Project Acceptance

This work is to be performed at Customer’s site. After receiving any release (the final/fixed release) during Acceptance Test duration, Customer will start doing acceptance immediately and last for maximum **2** weeks. So there may be more than one Acceptance Test cycle (in case the previous acceptance test cycle has been **unsuccessfully** **stopped** because of bugs) and the total Acceptance duration may be longer than/equal to **2** weeks.

The following are the collaborative approach**:**

1. Customer shall propose the acceptance test cases and the acceptance test cases need to be agreed by both Customer and GCS.
2. Customer shall execute the test cases on delivered system.
3. During the test execution, Customer will note and verify all test results and feedbacks to GCS.
4. GCS fixes found bugs with the agreed time.
5. GCS shall deliver updated software package to Customer for re-testing in 2-3 working days from date GCS receives bug list report. This is to ensure that GCS has sufficient time for regression testing.
6. The acceptance is considered done when an acceptance criteria (accepted number of bugs per bug level) agreed by both Customer and GCS is passed.

# Meetings, Briefings and Reports

### Meetings and Briefings

GCS will participate in and host the necessary reviews with the customer. Typically, the meetings will be held via conference call. However, GCS will be available to meet at the customer site, our facilities, or locations designated by the customer. We will take minutes at every review and distribute them no later than 5 days after the review.

### Progress Report

Progress reports will be delivered to the Customer weekly on each Friday that details project status, issues, or problems needing attention, etc. In addition we will be utilizing collaboration tools such as GotoMeeting, Skype, and Yahoo Messenger which allow for casual team/customer collaboration. Use of GotoMeeting will be provided free of charge during the development of the WDE System.

# Software Development Process

For the WDE System project, we will utilize the GCS Software Development Process (GCS SDP) which is developed and implemented based on GCS characteristics, development methodology and practice points gathered from ISO requirements, Rational Unified Process (RUP) and compliant with CMMI level 4 requirements.

GCS SDP is organized into five groups: Software Development Life Cycle (SDLC), Software Maintenance Life Cycle (SMLC), Project Management Life Cycle (PMLC), Project Supporting processes, and Organizational Processes (figure 4).

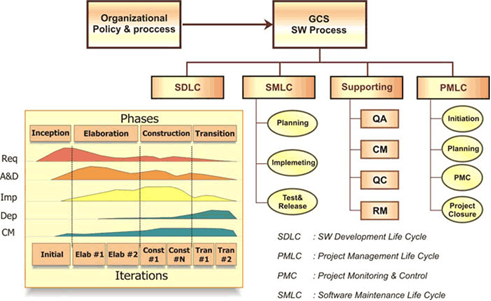


Figure 4 - GCS Software Development Process

The following diagram describes the GCS software development life cycle (SDLC) that is specific for developing secure software and will be applied in this project:

**Secure questions  
Abuse cases**

**Requirements**

**Arch & Design**

**Test plans**

**Implementation**

**Deployment**

**Maintenance**

**Threat & Risk  
analysis**

**A&D**

**Review**

**Group member**

**training**

**Risk based**

ystem MS and SEO) (x86 version databases**security test**

**Review old defects**

**Check-ins checked**

**Secure coding guidelines**

**Use tools**

**Security review**

**Learn &**

**Refine**

**Security push**

**Test**

**Security tests**

**Penetration tests**

Figure 5 – Secure Software Development Life Cycle

The details of each phase during the above development life cycle include:

|  |  |
| --- | --- |
| * Inception | * + Security questions   + Security requirements and abuse cases   + Security training for developers |
| * Elaboration | * + Secure architecture and design patterns   + Architecture and design security review |
| * Construction | * + Secure coding standards, checklists. Use GCS Security Coding Standard   + Code review, static analysis   + Security testing |
| * Transition | * + Final security review     - Remove sensitive comments, esp. in HTML that expose to user     - Remove default parameters, accounts, …     - Remove test scripts     - Protect admin control panel     - Code reviews of risky code     - Analyze incoming security bugs   + Learn and refine |

# Effort and Schedule

To be defined when Customer has detailed requirements and expected timeline.

# Assumptions

The following assumptions have been used in defining project development effort and schedule. If any of the assumptions are incorrect, GCS reserves the right to negotiate with Customer to either reduce the scope or change to the project’s price and schedule.

* The quoted effort and cost were estimated for the functionalities specified in scope of work. If there are additional requirements, GCS reserves the right to re-estimate the project effort and cost
* This proposal assumes that UI design and report templates are provided to GCS by Customer prior to project starts