**Architecture Pre-Review**

**Questionnarie: eFACTS**

***Volvo GTT, Complete Vehicle, Feature Verification&Validation***

**Document Version: <Version>**

**Distribution:**

<Name> <Function/Role/Project>

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 2015-05-25 | MS | Draft 0.1 |

Table of Contents

[1 System Description 6](#_Toc336599841)

[1.1 Business Case 6](#_Toc336599842)

[1.2 Business processes and main use cases 6](#_Toc336599843)

[1.3 Users 6](#_Toc336599844)

[1.4 Functional Area 6](#_Toc336599845)

[2 Architecture 7](#_Toc336599846)

[2.1 Conceptual description 7](#_Toc336599847)

[2.2 Logical description 7](#_Toc336599848)

[2.3 Bought package 7](#_Toc336599849)

[2.4 Servers 8](#_Toc336599850)

[2.5 Clients 8](#_Toc336599851)

[2.5.1 GUI Issues 9](#_Toc336599852)

[2.5.1.1 Common Look & Feel 9](#_Toc336599853)

[2.5.1.2 Internationalization 9](#_Toc336599854)

[2.5.1.3 Web Technology 9](#_Toc336599855)

[2.5.1.4 Mobile client 9](#_Toc336599856)

[2.6 Technology Platforms 10](#_Toc336599857)

[2.7 Databases 10](#_Toc336599858)

[2.8 Configuration and Extensibility 10](#_Toc336599859)

[2.9 Frameworks 10](#_Toc336599860)

[2.10 Technical components 10](#_Toc336599861)

[2.11 Re-used technical solutions 10](#_Toc336599862)

[2.12 Reporting tools 10](#_Toc336599863)

[2.13 Monitoring Tools 10](#_Toc336599864)

[2.14 Integration 11](#_Toc336599865)

[2.14.1 Internal Communication 11](#_Toc336599866)

[2.14.2 External Communication 11](#_Toc336599867)

[2.15 Information 11](#_Toc336599868)

[2.16 Security 11](#_Toc336599869)

[2.16.1 Authentication 11](#_Toc336599870)

[2.16.2 Authorization 11](#_Toc336599871)

[2.16.3 Encryption 12](#_Toc336599872)

[2.17 System Topology – Deployment view 12](#_Toc336599873)

[3 Development 13](#_Toc336599874)

[3.1 Software development method 13](#_Toc336599875)

[3.2 Code reviews 13](#_Toc336599876)

[3.3 Test approach 13](#_Toc336599877)

[3.4 Bug correction routines 13](#_Toc336599878)

[3.5 Build routines 13](#_Toc336599879)

[3.6 Code versioning 13](#_Toc336599880)

[4 Lifecycle Process 15](#_Toc336599881)

[4.1 Installation routines 15](#_Toc336599882)

[4.2 Release handling, patch handling principles 15](#_Toc336599883)

[4.3 Change request handling 15](#_Toc336599884)

[4.4 Change management 15](#_Toc336599885)

[4.5 Version management 15](#_Toc336599886)

[4.6 Guidelines and rules 15](#_Toc336599887)

[5 Known Issues 16](#_Toc336599888)

[5.1 Availability 16](#_Toc336599889)

[5.2 Modifiability 16](#_Toc336599890)

[5.3 Performance 16](#_Toc336599891)

[5.4 Security 16](#_Toc336599892)

[5.5 Testability 16](#_Toc336599893)

[5.6 Usability 16](#_Toc336599894)

Project/Solution Name

eFACTS – Electronic Fault Capturing System

Review Objectives

The aim of the review is to provide ITS with architecture status in order to get EFACTS application under ITS management.

Review Scope

Infrastructure review

Devices and network

OS, DB, Frameworks

Software architecture

Software integrations

Data flow

Technologies

Project/Solution Contact

BF59070, Martin Svennungsson, +46 31 3225845

Corporate Process and IT Contact

Antbäck Helena [helena.antback@volvo.com](mailto:helena.antback@volvo.com)

Review Steps

Pre-Review

This pre-review document should be filled in and returned no later than;

Date: 11/09/2015

Review

A review will be held;

Date: 18/09/2015

Location: Skype

Post-Review

NA for now.

Questions

# System Description

The system enables to capture faults found in different trucks and to distribute this data to the responsible solution owner at engineering. The data stored could be used to generate reports on the vehicle status for each SW release.

The system contains of two application eFACTS Collector and eFACTS Refiner.

Collector is used to do a binding between a logger, vehicle, test engineer and truck SW release, in order to upload data to the correct vehicle.

Refiner is used to analyze the logged data, adding more metadata and create a link to this recorded data in the PROTUS –system (using a url-link copy/paste).

## Business Case

Reduce lead time for handing faults found in trucks during testing and distribution of the data to engineering.

<http://violin.volvogroup.net/sites/news/vg/gt/gtt/Pages/Reducing-lead-time-from-2-weeks-to-10-minutes-.aspx>

Quality award

An article was published in Global Magazine page 18+19 (Quality award)

<http://dmweb.v-tab.se/webpages/Volvo_Global/VOGL-140006-B.html>

Link broken

## Business processes and main use cases

Support test engineers during Perform Verification & Validation at FVV/CV.

Feature Verification & Validation, Complete Vehicle

Deliver data and results to engineering within GTT.

## Users

There are 78 main users running the desktop applications, these are test engineers and measurement engineers.

The sites using the application are GOT, LYS, GSO, CUR, BLR, AGO, UK/AUS

Proving grounds: Hällered, Motegi, La Valbonne, Hoskote, Randon

Most of GTT engineers are using eFACTS recorded data from directly PROTUS. (Link to eFACTS is manually added in PROTUS report).

## Functional Area

Lead Organization: GTT

Lead Sub Organization: Vehicle Engineering

Yellow Pages MDID: 31160 – EFACTS

<http://yellowpages.volvo.net/InformationSystem/f3038d95-f7fc-4d57-8948-c49ec0e0d58f.asp>

# Architecture

This application has been developed internally without architecture design, but mainly based on satisfying incoming user requirements.

## Conceptual description

eFACTS Collector Desktop Client

* Creation of bindings between logger systems and vehicles (test objects)
* Uploading measurement data manually
  + Quick analysis to determine type of data
  + Split data into externals, autotraffic and raw
  + Upload split packages

Remote Data Collector

* Collects measurement data from a specific location on our disk using FTP, where RDM sends incoming measurements
  + Quick analysis to determine type of data
  + Split data into externals, autotraffic and raw
  + Upload split packages

Organizer

* Backup of original raw data
* Split autotraffic data into externals and raw data
* Analyze raw data to extract simple meta data for the measurements and included events
* Handle checked in and discarded events
* Deleting old raw data
* Sending statistics to end users by mail

eFACTS Data Refiner

* Simple analysis of raw data
  + Watch video
  + Listen to sound
  + See truck data at the time of the event
* Add data to events
  + Add values for predefined keys (mostly PVT specific data)
  + Connect events to a PROTUS report (in eFACTS)
    - Generate a http-link pointing to the event data, to paste into the PROTUS report
  + Ask for help using the ”Assist Me” functionality

Converter

* Creates ”Assist Me” downloads and deletes old ones
* Converts data paths to its final form

External Services

* Sends external data to the project
  + Currently only to the Safety File Server

Service Watchdog

* Periodically checks the state of our server side services and tries to restart them if they are not in a running state
* Checks the legacy watch dog file for services that has very long running operations (Organizer and Converter)

Truck Specification Importer

* Performs database backup
* Imports truck specifications from a flat file in a specified directory

eFACTS Externa Data Import

* Tool used for downloading external data from eFACTS and converting the data into different formats using third party application IPEconverter

PVT Manager

* Application for managing the PVT test code
* Also manages some of the data used in the rest of eFACTS

Engineering Report Tool

* Generates part of the ER-report for a PVT test

eFACTS Client Services

* Simple implementation of a WCF service to make database data available to client applications without storing sensitive data in the client (DB credentials)

Old WebForms Web Page

* Accessing data from PROTUS report links generated in the eFACTS Data Refiner
* Searching for measurement data

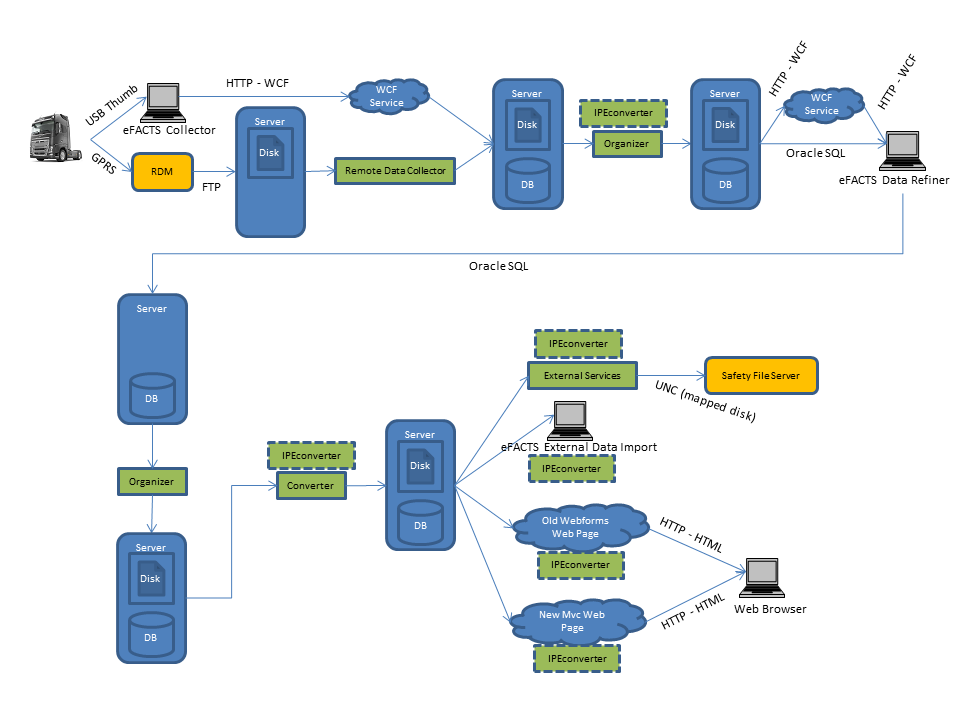
New MVC Web Page

* Currently a draft, quick solution for providing access to data from TeDM
* Used for accessing external data from the TeDM system

IPEconverter

* Third party application from IPEmotion used for converting measurement data from the MLog-systems

## Logical description



Logger data is transferred to eFACTS in two ways (upper left corner):

Manually via USB thumb

* + The logger transfer data to a USB thumb
  + The data is copied to the user computer
  + The user runs eFACTS Collector to upload data to the eFACTS Server
  + The data is handled by the Client Services (WCF)

Automatically via RDM

* + Data is uploaded to RDM via GPRS
  + RDM transfers data to eFACTS using FTP
  + The data is collected by the Remote Data Collector service on the eFACTS Server

The Collector applications split and categorize data in three different categories, which determines how they are to be treated:

* External data
  + Data related to external projects (not PVT)
  + Data stored in the EXTERNAL directory
  + Meta data stored in the Z\_EXTERNALS table
* Autotraffic
  + Data that contains auto-triggers
  + Data stored in the AUTOTRAFFIC directory
  + Meta data stored in the Z\_PENDING\_AUTOTRAFFIC table
* Raw
  + Raw data that match the PVT project
  + Data stored in the RAW folder
  + Meta data stored in the Z\_PENDING\_MEASUREMENTS table

The data is later handled by the Organizer service running on the server:

* Autotraffic
  + Is analyzed and split into External data and Raw data if applicable
* Raw
  + Is analyzed and information about each measurement event is stored in the Z\_PENDING\_EVENTS

After being handled by the Organizer, Raw data becomes available to the user in the eFACTS Data Refiner. The user can:

* Watch video and listen to sound recordings from each event
* See signals snapshot values for each event
* See GPS-coordinates on a map (Google Maps)
* Submit additional information about the fault
* Connect the log to a PROTUS report
* Mark the events for check in or discard them

Once all events of a Raw measurement has been either checked in or discarded, the Organizer will handle them. The discarded events are deleted and the checked-in events will be turned in to Data paths, as they are called.

The data paths contains essentially the same information as the Z\_PENDING\_EVENTS but are used as a more permanent storage.

The data is transferred to the DATA directory and the metadata to the Z\_DATA\_PATH table.

Once the data is in the Z\_DATA\_PATH table, the Converter service convert the data into a form more suited for downloads.

## Bought package

Not applicable. The application has been developed internally within FVV.

## Servers

PROD: SEGOTN11614 (Windows 2008R2)

TEST: SEGOTN11615 (Windows 2008R2)

## Clients

Webpage – access to data attached to PROTUS reports and supports general search within this data.

Windows Desktop Clients – Used to upload and refine the measurement data.

The installed applications today requires Open client and Oracle SQLNET client

Which client operating systems are used?

Windows 7

Does the solution include help functionality?

Yes, previously in the form of CHM (Compiled HTML) files included in the installations. CHM documentation is out dated.

Newly started Teamplace Wiki page is more up to to date:

<https://teamplace.volvo.com/sites/3p-EEE-CV/eFACTS%20wiki/Home.aspx>

Has an outside-in perspective been considered, a data-entered-once approach?

Outside-in has not been considered, but data-entered-once is used in some parts of the application.

### GUI Issues

#### Common Look & Feel

If it is a web GUI, does it comply with the Volvo Group Web Application Design Manual?

Which common look & feel is used?

Is the GUI changed compared to previous versions?

The web page GUI partly complies with Volvo Group design guidelines.

#### Internationalization

How is support for languages like Chinese?

Are language texts hard coded in the source code or separated?

Does this also include the help functionality?

Applications are in English, no support for other languages today.

Some data is translated into the languages of the sites (PVT)

#### Web Technology

Which web technology is used, web servers, GUI renderers, etc.?

WebForms, ASP.NET – Old PROTUS data download

.NET MVC – TeDM Access Web Page

IIS 7

#### Mobile client

NA

## Technology Platforms

Which technology platform is used, e.g. J2EE, .NET, IMS, etc.?

.NET 4.0

## Databases

List the used database systems.

Databases – Oracle 10g

GCVELP02 - gorapre9

GCVELT01 – gorabas11

GCVELP01 – orapre3 (to be closed)

## Configuration and Extensibility

How is the system configured?

How can you extend the system?

The applications are configured mainly in App.config files, today we apply different configurations for the environments during the builds of MSI packages using WIX XmlFile commands.

We also have some application and user specific configuration that is fetched from the database at runtime.

## Frameworks

List all used frameworks.

.Net (WinForms, WPF, WebForms, .NET MVC)

## Technical components

List technical components used, both in-house developed and third-party components.

Are open source components used?

Infragistics (license)

NHibernate

DotNetZip

AutoMapper

EnterpriseLibrary (Logging)

Unity

Prism

NPOI (Excel XLS creator)

Ajax Control Kit

Open XML

Oracle Data Access Client (recommended, but not mandatory)

## Re-used technical solutions

None

## Reporting tools

Which reporting tool is used?

Enterprise Library logging application block with customization.

## Monitoring Tools

How is the system monitored?

Are you using internal monitoring tools?

Is it possible to use Volvo standard monitoring?

The server applications are monitored partly by the Windows Task Scheduler and by a homemade service checking the time of the last log entry of each service, sending a mail to the support inbox if nothing has been logged in a long time.

## Integration

We’ll provide additional details after discussions about TeDM on Monday the 14th of September.

Waiting for the MQ client to be installed on the server.

Has the Integration Office been involved?

No

Are Integration Requests created?

Yes…

Describe the overall project integration approach.

N/A

### Internal Communication

How does the system communicate internally between components?

The application components communicate primarily by persisting data to the database and file storage that is then accessible from the other applications.

For database communication, the Oracle Data Client is used if installed, otherwise the .NET System.Data Oracle Client. Some of the database communication is now also done using a WCF service consumed by the desktop clients, to avoid having to store database passwords in the client and to improve performance.

### External Communication

How does the system communicate with external systems, i.e. broker, message bus, MQ, VCOM, etc.?

Is the Volvo Group Integration Platform used?

FTP for receiving data from RDM (Remote Data Management) and Truck Specifications

UNC-path (mapped drive) to Safety Logging Data server to upload data

For PVT use (tightly connected to eFACTS): Http client for parsing the PROTUS report pages (HTML), to be able to cache important information.

We will use MQFTE for integration with the TeDM (Test Data Management) system.

## Information

Have you secured you get the data from the correct sources?

Yes.

RDM user pass

EDB

PROTUS

## Security

### Authentication

Is authentication used? If so, which solution is used?

The Data Refiner does require the user to enter username and password, which are validated using the VCN Active Directory.

The system is only accessible within VCN.

### Authorization

Is authorization used? If so, which solution is used?

A database table (Z\_CVEL\_USERS) containing users and their roles, i.e. role based authorization.

Check towards Z\_CVEL\_USERS and Active Directory for “eFACTS Refiner” application

No authorization for “eFACTS Collector” application

T0g2780 user id is used to map the disk towards the external Safety Logging Data server.

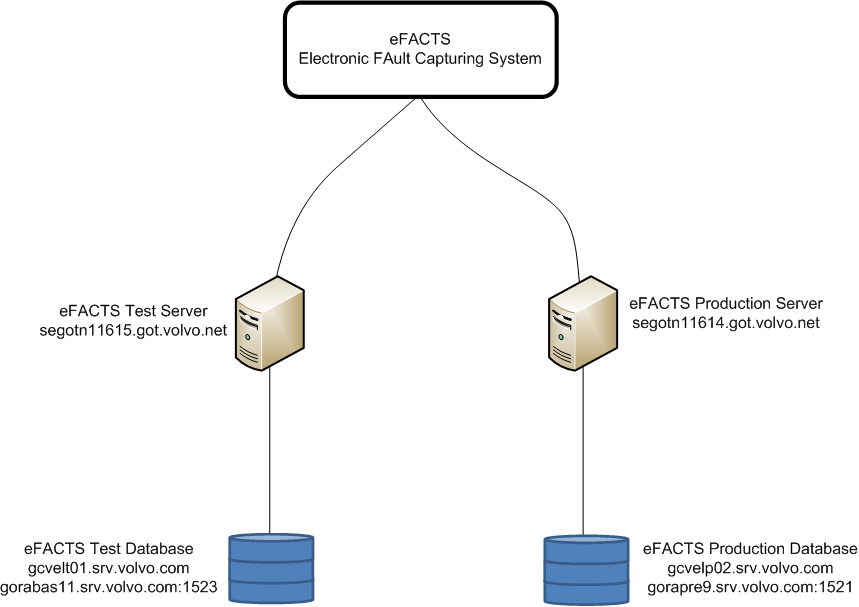
### Encryption

Is encryption used? If so, which solution is used?

None, not required since only accessible within VCN (internal information)

## System Topology – Deployment view

Describe the topology of the system (deployment view).



# Development

An eFACTS folder exists on the related PVT Teamplace:  
[https://teamplace.volvo.com/sites/3p-EEE-CV/PVT%20Team%20documents/Forms/AllItems.aspx?RootFolder=%2Fsites%2F3p-EEE-CV%2FPVT%20Team%20documents%2FEFACTS&FolderCTID=0x01200016F9A74E1446794A91E43FC95B58460E&View={5257F430-130A-4479-9819-5AAD2A57AE20}](https://teamplace.volvo.com/sites/3p-EEE-CV/PVT%20Team%20documents/Forms/AllItems.aspx?RootFolder=%2Fsites%2F3p-EEE-CV%2FPVT%20Team%20documents%2FEFACTS&FolderCTID=0x01200016F9A74E1446794A91E43FC95B58460E&View=%7b5257F430-130A-4479-9819-5AAD2A57AE20%7d)

## Software development method

Which software development method is used, e.g. agile?

Agile and informal sprints with small team

## Code reviews

Do you perform internal code reviews?

No

## Test approach

Describe the test approach and test routines.

Do you use unit tests? Are they automated?

Architecture doesn’t support unit test at the moment.

First tested on TEST server and the pushed to PROD, but tested by releasing to smaller groups of super users first e.g. using client Release Candidate against production.

## Bug correction routines

Are there bug correction routines in place?

Is a unit test created and bug reproduced prior to fixing the bug?

Reporting from users to “Support E-FACTS” mailbox.

Discussed with users to reproduce the bug.

Usually then writing a bug work item in TFS, testing if major changes.

(Control by releasing to smaller super users first e.g. using Release Candidates)

We have added exception handling routines in the applications, fetching all unhandled exceptions, giving the user a possibility to report the error to the support including a personal message, the current runtime call stack and the application configuration values.

## Build routines

Is Daily Build used? Describe the build routines.

Built by developer on local machine.

## Code versioning

Describe how the code is version handled.

Using Team Foundation Server:

<http://tfs.it.volvo.net:8080/tfs/Global3/SEGOT-EFACTS>

No labeling of versions

# Lifecycle Process

## Installation routines

Describe the installation routines for the solution.

Applications are installed using MSI installation packages outside of FAROS, distributed via common disks or download from WCF service (auto updater).

## Release handling, patch handling principles

Describe if there are routines in place and what they look like.

Z\_GENERAL has information about which is the latest version of the particular application used. If the client has a lower version number the application might suggest or force the user to update the application. The new installer is downloaded via a WCF service and started automatically.

* Version numbers
  + The version number used by eFACTS should be in the format <major>.<minor>.<build>.<revision> where the revision part is always zero when published to customers
* Distribution of installers
  + The installers are placed in a directory on the application servers and a value in the configuration table (Z\_GENERAL) of the database is updated to indicate that the new version exists.  
    When the user starts the application, the application will compare the current version with the version number available in the database. If the one in the database is higher, the user will be forced to update or to quit the application. I the user chose to update, the latest MSI package will be downloaded through a WCF service and started automatically.

## Change request handling

Describe if there are routines in place and what they look like.

Error reports from users to the “Support E-FACTS” mailbox. The mails are often followed by a short discussion with the user to reproduce a bug. Before the job is handled, often a work item is added in the TFS.

## Change management

Describe if there are routines in place and what they look like.

Do you have any train-the-trainer concept in place?

We have weekly meetings with the development team and also with key users.

## Version management

Describe if there are routines in place and what they look like.

Are the separate services versioned or is entire application versioned?

Separate applications and services are versioned independently. Of course there are some dependencies between the applications which might make it necessary to update two or more applications at the same time.

## Guidelines and rules

Describe if there are routines in place and what they look like.

No guidelines have been appointed for the development.

# Known Issues

List all known issues in the following areas.

Most known issues and change requests will be available in the TFS.

The progress is slow, because most of the applications are written to work with DataSet and SQL, almost everywhere. Also, these datasets are sometimes persisted into database CLOBS directly and sometimes bound to in user interface components (Infragistics) directly. This makes the progress much harder to perform without “changing everything at once”, i.e. rewriting the applications. And rewriting the applications has been put on hold, since we then might as well redesign the entire data model and database structure to better fit the work we are trying to perform.

## Availability

Are there any issues concerned with system failure and its consequences?

Lately a lot of bugs have been fixed making the system much more stable, so there are no current known availability issues except for server maintenance work that sometimes requires the server side applications to be shut down.

## Modifiability

Are there any issues with the cost and time and ease of change?

Yes, since the applications architecture was inherited from a prototype system which continually have been growing, there is a lack of use of common coding patterns, guidelines and practices within the code. This has resulted in increased time for implementing new features and fixing errors. Mainly the maintainability issues are caused by dependencies in the system, without any compile time help, e.g. SQL queries in strings, and business logic is mixed with data access and UI code in the user interface. A lot of code is written specifically for third party components and ADO data sets.

## Performance

Are there any performance problems within areas such as database access, messaging, system response, etc.?

The access to the database and server is very chatty and mostly synchronous, which leads to performance problems on some remote sites.

Data is also converted multiple times, even when not necessary. Data is stored in ZIP archives that have to be extracted and re-compressed for every single analysis or conversion. This causes very heavy system load.

We use the DotNetZip library for extraction and compression of ZIP archives. We saw in some version(s) of the library that the process would sometimes hang when running in the default multithreaded mode. This is a known issue, that has not been properly fixed and therefore we are running all ZIP processes in single threads instead causing some applications and services to slow down.

CR -1431 for slow performance for sites, especially outside Europe.

## Security

Is there a risk of unauthorized usage of the system?

We have not applied any additional security measures except for limiting access to VCN users/network. There are database credentials stored within code and sometimes FTP credentials are exposed within links visible to the user.

We started to move all applications to using WCF instead of SQL clients to make the system more secure and robust. But there have been many more important issues, e.g. fixing critical bugs so the progress has not been finished fully.

## Testability

Are there any problems with the test method being used? Does it detect faults to an acceptable degree?

Yes, since we rely on users testing the applications a lot of bugs are found running the applications towards the production environment.

## Usability

Are there any problems with the usability of the system?

A lot of business and data persistence logic runs within the UI threads of the desktop applications. This might lock the UI from time to time.

Sometimes users are not familiar with the terminology used in the user interface.

A lot of data is presented in Excel like data grids, with SQL like filtering. It might be hard for users to get a good overview.

Drag and drop behaviors and similar functionality are built as a mix of user interface code and business logic 🡪 Hard to maintain and change without breaking something.

# Review Outcome and Proposed Actions

The DRS Architecture Review is an architectural review focusing on identifying issues, ranking them and proposing actions to take and also describing consequences if no action is taken. The DRS Architecture Review is looking at the situation at the time of the review. The purpose is to identify issues and propose actions forward. The identified issues and proposed actions must be seen in the context of the architectural status at the time of the review.

## Architectural Issues and Volvo Group Target Architecture non compliance

### System architecture

#### Tightly coupled with other systems

**HIGH**

|  |
| --- |
| **HIGH** |

* PVT, PVT has direct access to eFacts database which is prohibited as any changes in eFacts DB will cause disruption of the communication between eFact and PVT.

**Recommendation:**

Use of an appointed integration (MQ).

#### Internal architecture

|  |
| --- |
| **HIGH** |

* eFacts client connect directly to DB which is prohibited, it should use dedicated interfaces
* UI implementation is tightly coupled with the database (use of .Net Dataset and SQL queries stored as strings)🡺 Very difficult maintenance
* Business logic/UI/and database access are not separated in different layers)🡺 Very difficult maintenance

**Recommendation:**

Refactor code using layer architecture in order to separate UI, Business Logic and data access.

#### Use case driven architecture instead of Domain Driven Architecture

|  |
| --- |
| **MEDIUM** |

Data conversions done in multiple locations in the system.

**Recommendation:**

Define Domains with clear responsibility and boundaries ensuring no overlap in between domains.

Refactor and centralize data management algorithms to avoid code duplicate and optimize data extraction flows.

#### Database

**HIGH**

|  |
| --- |
| **HIGH** |

* Use of Oracle DB, not appointed in VIAP, this might cause interoperability issues

**Recommendation:**

Migrate to Miscrosoft SQL Server.

#### Third party components

**HIGH**

|  |
| --- |
| **LOW** |

* Infragistics (with license) not appointed in VIAP should be Telerik
* Enterprise Library (logging), should use Volvo NVS

**Recommendation:**

Use of appointed third party components.

#### Integrations

**HIGH**

|  |
| --- |
| **HIGH** |

* Use of FTP in various locations which is prohibited
  + RDM 🡺 eFacts
  + Truck Specification 🡺 eFacts
  + eFacts 🡺 eFacts external Data Import
* PROTUS report pages, direct HTML parsing of data in eFacts to retrieve data from PROTUS report. Any changes in the PROTUS HTML formatting may cause a system disruption.

**Recommendation:**

Use of VG appointed integration technologies

#### Security

eFacts is dealing with internal classified information.

**HIGH**

|  |
| --- |
| **MEDIUM** |

* No use of SSO (Single Sing On), user has to typing in their credentials each time they use the application
* Application is opened to all VCN users, no restriction supported.
* No use of VCD to register users, it’s done locally within the application. 🡺 Has to use VCD for identity management (Authentication, Authorization)
* Custom business roles, ok to be managed within the application.

Volvo Group Guidance for .Net application detail can be found here: <https://teamplace.volvo.com/sites/volvoit-dotNET/Web%20client%20wiki/Security%20in%20NVS.aspx>

**Recommendation:**

Use of VG appointed security features.

#### Data Treatment:

**HIGH**

|  |
| --- |
| **MEDIUM** |

Useless and resource consuming zip file management of Audio/Video files which are already compressed, this has caused system overload causing system disruption.

The mass storage structure not optimized, creating complex algorithms to find the right data

**Recommendation:**

Review media file data flow and remove useless zipping/unzipping operation.

Review Mass Storage structure.

### Stability

**HIGH**

|  |
| --- |
| **High** |

Stability issues occur during high system load, and occur 1 to 2 times a month.

* Instability might cause manual intervention to restart the server system
* In case of system error manual intervention to repair the system and cleanse database by an expert might be required.

**Recommendation:**

Perform a stability detailed analysis in order to define an action plan.

### Release management

#### Distribution

**HIGH**

|  |
| --- |
| **HIGH** |

* The new tool releases, MSI packages, are distributed using a shared folder, and update is done on user willingness
  + No guarantee of the user has the latest version of the client, which can cause some version compatibility issues

**Recommendation:**

Client application deployment should be done using Faros, or at least a ClickOnce package.

#### Version Management

**HIGH**

|  |
| --- |
| **HIGH** |

No labeling in source management system

No compatibility table between different components of the application

**Recommendation:**

Manage proper version control using labeling in source configuration tool.

Keep track of the compatibility table in between different modules of the application.

### Change Management

**HIGH**

|  |
| --- |
| **HIGH** |

* Proactively managed by developer using a support mailbox, no tracking tool is used.
* End user acceptance test done in production environment by selected key users.

**Recommendation:**

Use of a CR/Deffect tracking tool (Remedy)

Perform user acceptance test on Test environment.

### Documentation

**HIGH**

|  |
| --- |
| **MEDIUM** |

* End User documentation out dated

**Recommendation:**

Upgrade documention