# Overall Project

This section of the Project Health Check document is an assessment of the projects overall status from a technical perspective.

## General Project Topics

* Is there an automated build or continuous integration process deployed?
* Will a single build support all environments?
* Are design reviews being conducted that include representation from the business and architecture to ensure that both the functional and technical objectives are being met?
* Are team or peer code reviews being conducted?
* Has a unit testing strategy been defined and implemented?
* Has a process been defined to troubleshoot infrastructure and application issues?
* Technical Project Staff (Configuration, Integration, Infrastructure, and Conversion)
* Is work performed with appropriate quality?
* Are the integration, infrastructure, and conversion team members allocated to the implementation at the appropriate skill level for an implementation of this complexity?
* Is an appropriate amount of mentoring, coaching, and knowledge transfer being performed by the onsite Guidewire team to the customer? Is there time allocated to allow this to happen?
* Is the team being used in an appropriate manner?

## Infrastructure

* This section of the Project Health Check document outlines areas such as architecture fire proofing, general environment setup, and performance testing.
* A large portion of the health check revolves around the recording of the platform in a document called the detailed platform definition, which is maintained by Guidewire to keep detailed information about a customer's environment.

### Clustering

* Do you use Cisco switches and, if so, are you aware of the configuration steps needed to let UDP/Multicast traffic flow across switches?
* Do you expect to run various application servers across multiple data centers? If so, does your network topology allow for UDP/Multicast traffic to flow across data centers?

### Virtualization

* Do you expect to use virtualization on one of the tiers (client, web, application server or database)?
* Have you specifically run full load performance tests with virtualization enabled?
* Do you overcommit CPU and/or memory on production environments (client, application server or database)?
* Do you live migrate production environments?

## Environment Status

This section lists the environments to be setup, what the planned completion time is, and any other comments.

* **<Environment Name>**

## Infrastructure

This section of the Project Health Check document outlines areas such as architecture fire proofing, general environment setup, and performance testing.

A large portion of the health check revolves around the recording of the platform in a document called the detailed platform definition, which is maintained by Guidewire to keep detailed information about a customer's environment.

### Performance Testing

#### Load

* Will/did performance tests include the same number of active users as expected during production? The concern is that many customers will run the performance with much fewer but more active users, which represents a less adequate model.
* Will/did you simulate a full day of load, as short performance tests may not identify certain issues, like memory leaks?

#### Data

* Will/did you generate data that is similar to what production should look like?

### Performance Test/Production Environment Monitoring

#### General

* Availability of a performance logging capability to track response times (server side at a minimum)

#### Application server

* OS level metrics (CPU, Memory) monitoring and analysis
* Garbage collection logs monitoring and analysis
* Cache utilization monitoring and analysis

#### Database server

* OS level metrics (CPU, Memory) monitoring and analysis
* Storage metrics (read and write load, response times) on the disks/luns supporting the database data
* Regular collection of reports (AWR/Statspack for Oracle DMV for MS-SQL). For Oracle, the reports must be collected both through Oracle and the server tools
* For Oracle, statistics collection and analysis through the server tools (no statistics should be collected on message tables)

## Execution - Integration A

This section of the Project Health Check document is an assessment of a single integration point.

### Design Topics

* Data Definition: Have you identified the data that needs to be sent?
* Message Format: Has the format of the data to be transmitted been identified?
* Message Verbosity: How much data is being sent and is it understood whether it is all necessary?
* Communication Protocol: Has the communication protocol for the data transmission been identified?
* Message Conditions: Are the conditions surrounding when data needs to be transmitted or received been defined?
* Data Transmission: Will the data be sent and processed in a synchronous or asynchronous fashion?
* Business Logic Application: Have you defined the business logic that needs to be configured in the Guidewire application as a result of the data transmission?
* Error Conditions: Are the error conditions properly documented with handling strategies? For example, in the case of Event Messaging based integrations, would the error cause destination suspension, retriable errors, or non-retriable errors?
* Error Recovery: Are error recovery mechanisms identified and documented (e.g. message re-sync, or batch re-runs, etc.)?
* Balancing: Are there processes in place to ensure that data are transmitted successfully from one system to another? This is not always applicable (e.g. a synchronous integration such as Policy Search).
* Resource Definition: Have external resources such as connections, queues, etc. been defined in a way so that changes to those definitions are the least intrusive possible?
* Concurrency: Have any concurrency considerations been addressed?
* Clustering: How will the integration behave in a clustered or load balanced environment?
* Integration User Definition: Are there any special integration users that need to be identified and have they been configured and granted the correct permissions?

### Source Code Review

* Event Message Rule Structure: Are Event Messaging rules following best practices as defined in the Integration Guide?
* Message Payload Creation: Are message payloads being built in the event message rules?
* Message Tracking: Is sufficient data being tracked on the message to identify its state for retry and recovery purposes?
* Custom Batch Process: Are Custom Batch Processes following best practices as defined in the Integration Guide?
* API Usage: Are GW APIs, both GScript\Java and SOAP, being used appropriately?
* SOAP API Aggregation: Are multiple SOAP calls being aggregated into a single custom GScript Web Service API when possible to reduce overhead?

### Error Handling

* Error Handling: Are errors and exceptions being logged and handled appropriately?
* Concurrency: Is code thread safe? Are instance variables only being set in the setParameters method of plugin implementations, etc.?
* Resource Allocation: Are resources such as connections, queues, etc. being allocated appropriately and cleaned up or closed when no longer in use?

## Data Migration

This section provides a high-level assessment of conversion activities and will help determine whether a consultation with conversion specialists is warranted.

* Data Scope Definition: What data are being migrated (e.g. open claims only, agency/direct bill data)? How much historical data are included?
* Which systems currently store this data?
* Data Migration Timing: Is it on the critical path to go-live? Does it include rolling conversion after go-live? What configuration changes were made to enable the approach?
* Data Volume: Number of data sources and the number of entities being migrated?
* Data Quality: How much required Guidewire application data is missing in the legacy data? How much legacy data is in the wrong format or type?
* Does data need to be cleansed before migration?
* Data Migration Tools: Is the migration toolset sufficient for the data to be converted and is the team sufficiently familiar with those tools?
* Data Migration Performance: Are there any specific data migration timing windows that need to be considered (e.g. conversion must take less than 12 hour window on the weekend due to system processing schedules)
* Staffing: Have staff been allocated for data migration? Has there been any Guidewire application migration-specific training?
* Acceptance Criteria: Has clear acceptance criteria been defined for the migrated data? This should include clear direction of precision and tolerances for data.
* Test Planning: Do existing test plans take into account variations that could result from the introduction of migrated data? Are there specific test plans to exercise migrated data? Have staff been allocated for data migration testing?

## Configuration

### Unit Testing Topics

* GUnit testing: Is GUnit testing being fully utilized?
* Testing Quality: Are unit tests performed with "real life" users and roles and not with SuperUser to insure code functions properly from an end user perspective?

### PCF Coding Topics

* Find expression usage: Are find expressions used with a priority on performance and user experience? Is filtering done in the find call, instead of on the subsequent query result?
* Modal PCFs: Are modal PCFs being used where appropriate?
* PCF variables: Are variables being used appropriately, especially for inputs into attributes like editable, visible, available, etc.?
* DetailView and ListView reuse: Are common DetailViews and ListViews being reused where appropriate?
* Triggersvalidation usage: Is there excessive use of triggersvalidation in PCFs? Excessive use of this feature could have a performance impact.
* Formatting: Do PCF variables start with a lower-case letter, especially if their name matches an Entity name?
* PCF embedded code usage: Is the embedded code used where appropriate? Is it overused? If embedded code exists, can the code be moved to a class or library?
* postOnChange/field trigger usage: Are field triggers being used as a replacement for postOnChange where necessary? Is postOnChange used sparingly?
* Expensive calls: Are expensive server calls cached in local variables?

### Data Model Coding Topics

* Data Model Controls: Are changes to extensions.xml being maintained and controlled centrally so that maximum data reuse and coherence is achieved?
* Data Model Controls: Are the extensions and custom entities clearly defined and balanced between performance and proper normalization?
* Naming Conventions: Are entity names coded following best practices?
* Entity Supertype and Subtype usage: Are subtypes being used appropriately for similar objects? Are fields being added to the appropriate type? Common fields should be added at the highest type possible so that the subtypes inherit the same fields.
* Data Model Definitions: Are labels reused in PCFs or do new extensions have different labels which could cause confusion with understanding the data model?

### Rule and Class Function Coding Topics

* Naming Conventions: Are rule names following best practices and are they named consistently?
* Error Logging: Are rule errors and messages being sent to log files?
* Loop expression usage: Do break or return statements exist in the loop when only one object needs to be found or updated?
* Commenting: Is commenting used effectively to describe rule and class function logic?
* Dot notation: Are there frequent usages of three or more levels of dot notation?
* Formatting: Is a consistent code style used across all configuration code, including bracket placing, variable naming, etc.?
* Avoiding Null Pointer exceptions: Do statements that refer to subbeans check for null prior to executing the statement?
* Unused Code: Is unused code removed so that the overall coding is not cluttered with inactive statements?
* Code Reuse: Are existing class extensions and enhancements reviewed for re-use before creating new class functions?