UNIVERSITY OF BEIRA INTERIOR
Engineering

Core Solutions Transformation

Filipe Miguel Carrão Gonçalves

Submitted to the University of Beira Interior in candidature for the Degree of Master of Science in

Computer Science Engineering

Supervised by Nuno Manuel Garcia dos Santos, Ph.D

Covilhã, September 2015

Acknowledgements

I'm thankful for the ones who were around and did not let me give up on this challenge. The ones always available for a needed motivation and inspiration. Being those my family or friends.
When writing this dissertation, I got in touch with some exceptional people. I always find it important to take the best of each with me, and improve myself with such examples.

Jorge Ferreira, thank you for all the help, time, and guidance.

Jorge Tavares, thank you for your correctness and assertive management.

Professor Nuno Garcia, thank you for your motivation, advice and support.

For my family and friends, I'm grateful for the kind words and the courage you gave me.

This goes to all of you.

Summary

This dissertation focuses on the description of the work carried out to perform the Core Solution Transformation on a major insurance company. The problem is that the systems in use are getting outdated and can't handle or process the amount of information being generated. This represents a blocking factor for natural business evolution and new strategies.

This dissertation work was done in a working context while at one of the "Big Four". They provide services for a variety of industries such as the Financial Industry, specializing in Core Solutions. It was in this company that the foundation to carry this work was found.

For the Core Solution Transformation, Guidewire was used as the main tool. It provides an extendable Core for Insurance Services, organizing the basic concepts of the industry and can be fully customized.

Keywords

Core Solutions Transformation, Insurance, Guidewire

Contents

1 Introduction **1**
1.1 Objectives . . . 1
1.2 Document Structure . . . 1

2 State Of The Art **3**
2.1 Insurance Industry Contextualization . . . 3
2.1.1 Claims Life-Cycle . . . 6
2.2 Financial Services Core Transformation . . . 8
2.2.1 TIA - The Insurance Application . . . 8
2.2.2 Guidewire . . . 12
2.2.3 Conclusion . . . 17

3 Project Management **19**
3.1 Project Releases and Increments . . . 19
3.2 Agile . . . 20
3.3 Conclusion . . . 21

4 Task 1 - Overpayment Write-Offs **23**
4.1 Overpayment Overview . . . 23
4.2 Task Objectives . . . 23
4.3 Solution Developed . . . 24
4.4 Conclusion . . . 24

5 Task 2 - Capitalizations **27**
5.1 Capitalization Overview . . . 27
5.2 Task Objectives . . . 27
5.3 Solution Developed . . . 29
5.4 Conclusion . . . 31

6 Task 3 - Manual Check Overlimit **33**
6.1 Manual Check Overlimit Overview . . . 33
6.2 Task Objectives . . . 33
6.3 Solution Developed . . . 34
6.4 Conclusion . . . 34

7 Task 4 - HATS - Legacy Integration **35**
7.1 Legacy Integration Overview . . . 35
7.2 Task Objectives . . . 36
7.3 Solution Developed . . . 36
7.4 Conclusion . . . 38

8 Conclusion **39**

Bibliografia **41**

List of Figures

2.1 Insurance topics overview . . . 3
2.2 Policy effective dates - work specific. . . . 4
2.3 Claim Workflow . . . 6
2.4 TIA - Claim Modules . . . 11
2.5 Guidewire Architecture . . . 13
2.6 Guidewire configuration technology - Three tier . . . 15
2.7 Claim maturity levels on ClaimCenter. . . . 16
2.8 Business and functional claim processes . . . 18

3.1 Agile organization at the project. . . . 20

5.1 Capitalization Class Overview . . . 31
5.2 Capitalization workflow overview. . . . 32

6.1 Manual check overlimit workflow. . . . 34

7.1 Web solution display. . . . 36
7.2 HATS - Solution Architecture. . . . 37
7.3 HATS - Solution Deployment . . . 37

List of Tables

2.1 TIA Modules Overview. 10
2.2 TIA Claim Modules Overview. 11
2.3 InsuraceSuite products overview. 13
2.4 Gosu actions on different working areas. 16

4.1 Overpayment status conditions. 24

5.1 Capitalization specifications across different benefit types. 28
5.2 Capitalization Table - line example. 28

List of Algorithms

1 Write-Off Overpayment Amounts 25
2 Create Net Recovery 25
3 Authority Limits 25
4 WageLoss get Capitalization Factor 29
5 WageLoss Adjust Capitalization 30
6 WageLoss Create Capitalization 30
7 NEL get Capitalization Factor 30
8 NEL Adjust Capitalization 30
9 NEL Create Capitalization 30
10 Survivor Create Capitalization 30

Acronyms

ADF	Application Development Framework
BA	Business Analyst
EAR	Enterprise Application Archive
FEL	Future Economical Loss
FNOL	First Notice of Loss
FROI	First Report of Injury
FSI	Financial Services Idustry
HATS	Host Access Transformation Service
JVM	Java Virtual Machine
NEL	Non Economical Loss
NFA	No Further Action
RAD	Rational Application Developer
TIA	The Insurance Application
UBI	University of Beira Interior
WAR	Web Application Archive
WAS	Websphere Application Server

Chapter 1

Introduction

The world is facing an information revolution. Data is growing exponentially as there are more and more information to be stored and processed. Companies are adapting and trying to keep up with such transformation.

There is a high demand for high ended tools capable of handling such amount of information, transform the obsolete technology behind it and develop systems capable of lasting long.

In the Financial Services Industry, the solutions presented are as generic as possible. This technology is supposed to be configurable for each business model of each company by continuous interaction between clients, business analysts and developers.

1.1 Objectives

The student's objectives are as follows:

- To achieve certification on the required software;
- To apply the best solution he can for every problem he finds;
- To develop a critical thinking towards all the challenges and tasks;
- To gather a deep knowledge on programming and Guidewire platform configuration;
- To gather a deep knowledge of Insurance Industry in order to better apply his knowledge on his tasks.

1.2 Document Structure

This dissertation is organized in six main chapters. As the first one is supposed to give a general overview of the project and the second one to state the project strategies and management, the last four are supposed to give some in-depth of the tasks engaged by the student.

For the State of The Art chapter, an introduction to the business and Core Solutions Transformations will be made. Some key concepts will be clarified, some tools used on the field will be presented and some context of the specific area the student worked on will be clarified. It will scope the readers attention to the foundation needed for the next chapters.

It is important to introduce all entities involved in such project, the state it is in, how it is organized and what development methodologies are used. This topic will be addressed in chapter 3: Project Management.

For the last four chapters, it will be described the tasks the student has been involved with, the task's objectives, development and conclusions.

Through whole this document schemes, screen captures and code reviews will be available in order to complement and clarify the ideas expressed.

Chapter 2

State Of The Art

As the student will engage in an Insurance Industry related project, the first section of this chapter will describe it briefly. the main objective of this document is to contextualize the technical work with the business.

The main challenges of the Insurance Industry technology will be discussed. Major software available will be presented and compared. The chapter will then focus on the platform used in this project, as giving some knowledge on it.

2.1 Insurance Industry Contextualization

This chapter is intended to be an introductory reading to the Insurance Industry business. In order to relate all the concepts and provide a better understanding of the topic, figure 2.1 provides a general overview of them all and how they are hierarchically organized between each other.

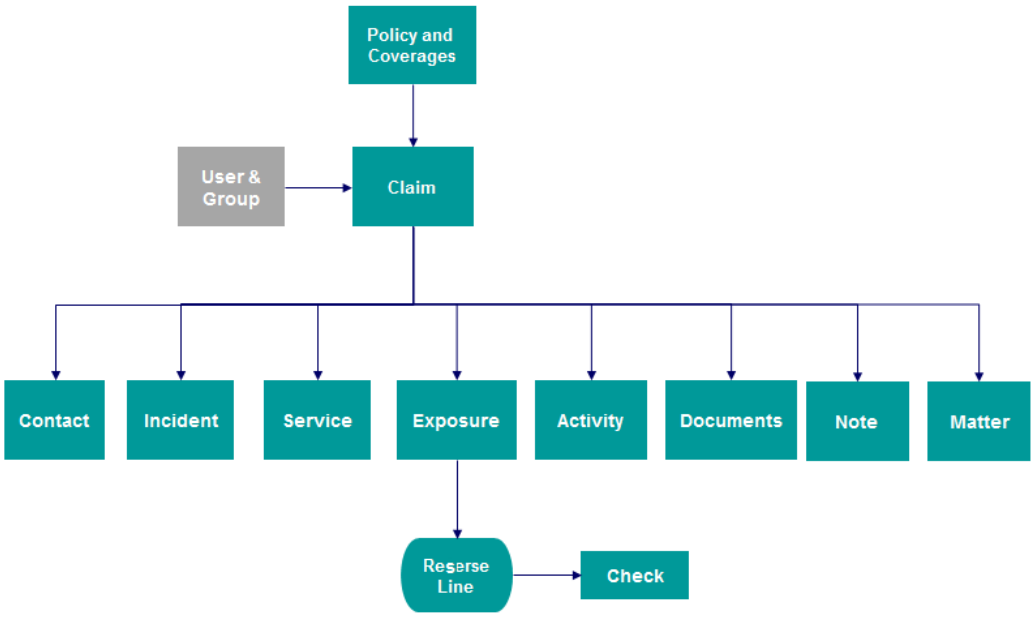

Figure 2.1: Insurance topics overview

A policy is a contract between the carrier and the insured in which the carrier promises to cover the insured for specific types of losses. A coverage is a type of loss that is or can be listed on a policy which the carrier will cover.

The term "insured" refers to the person, group of people, or business covered on the policy. The

term "third party" refers to a person who suffers a loss for which the insured is responsible.

A property coverage is a coverage in which a tangible asset (a physical object, a real estate location, or the body) of the insured is covered. The coverage exists to repair or replace the asset if it is lost, damaged, or otherwise rendered unusable.
A medical payment coverage covers medical payments (damage done to the insured's body). When a claim is filed, the money from losses covered by property coverage goes to the insured. (In some cases, it may go to a business which provided services to the insured, such as an auto shop which repaired the insured's car). But from a logical standpoint, the money is still going to the insured.
A liability coverage is an coverage in which the liability of the insured is covered. The coverage exists to provide financial remuneration if a third party suffers a loss for which the insured is liable. When a claim is filed, the money from losses covered by liability coverages goes to the third party, which is owed money.

A coverage applies for the claim depending on:

- Policy effective date
- Exclusions on the policy (exceptions)
- Terms such as limits and deductibles (earnings)

In this project in specific the Policy effective date is divided as represented on figure 2.2, for each effective date an accident may be on, different coverage types are applied.

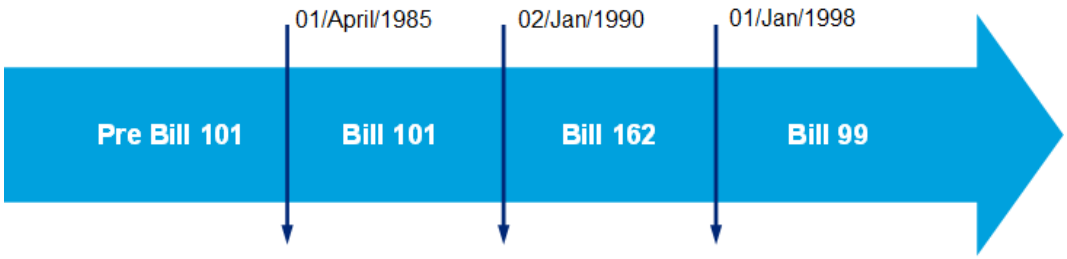

Figure 2.2: Policy effective dates - work specific.

Initially, a claim can be described as an event in which potentially covered losses occurred. One aspect of the claims process is to reference the relevant policy to ensure that the loss was in fact covered by the terms being. Even if there is a coverage on the policy, the loss may not be covered because:

- It occurred when the policy was not in effect.
- It was a kind of loss specifically excluded from a policy (such as a homeowner's policy which specifically excludes earthquake damage).
- The amount of loss is outside of the terms of the coverage.

Contacts are typically captured when the claim is first created. Some information about contacts may need to be gathered later. Each contact involved in a claim may have more than one role, such as a reporter, claimant, witness, doctor, attorney, repair shop, and so on.

The one current case where an incident does not represent the damaged item is the "Living Expenses" incident - pensions. The incident could belong to the insured (which is covered by a property coverage), or belong to a third party (which would be covered by a liability coverage).
Incidents are typically captured when the claim is first created. Some information about incidents may need to be gathered later, but the most ideal circumstance involves an intake process in which all the information about the "what happened" is captured.
Service requests are submitted as soon as possible. The service request can be for a quote to compare prices, a service to be performed, or for both a quote and service. Typically, a Service is tied to an incident such as requesting locksmith services for a burglarized home, or requesting vehicle damage assessment for one or more damaged vehicles, or requesting towing for a disabled vehicle. Each incident can have one or many services.

For many claims, a single occurrence of loss involves more than one coverage, and more than one claimant. Exposures are the mechanism used for tracking the progress of each possible indemnity - an indemnity is a payment in compensation for a loss. In property and casualty insurance, the goal is to restore a policyholder to the same financial position after the loss as he was in prior to the loss, without allowing the policyholder to profit from the loss.

Every activity is ultimately assigned to a user, who is responsible for completing the task. The activity identifies when the task is expected to be done (due date) and whether it has been done or not (status). Theoretically, one could say the process of processing a claim is identical to the process of completing its activities. Legacy claim systems often have a "diary" feature. This feature is used to manage claim-handling activities. A user can indicate that he or she has to look at a claim on a particular day. The diary feature is often limited, particularly in terms of how much text you can enter and how the visibility of this text is controlled, the number of references a given user can make for a claim on a given day (often limited to one), and the logic behind how diary activities get assigned , see section 2.2 for more Legacy information. Activities let both users and the system create and assign tasks.
Notes let users record information about a claim and associate it to the relevant part of the claim (such as the claim itself, a contact on the claim, or an exposure) without having to tie the note to a specific date or user.

A document is an electronic file or physical piece of paper which contains information relevant to the claim, such as:

- First Report of Injury
- Notification of Pending Inspection
- Affidavit of Vehicle Theft

Documents can be used to:

- Track information on physical pieces of paper (such as photographs of a damaged car, diagrams of the floor plan for a workplace, assessments from mechanics or diagnoses from a doctor)
- Track information that exists electronically (such as email correspondences, word processing documents that were mailed to the insured, and scanned photographs of property that was stolen)

A reserve line can be thought of as a amount of money set aside for one specific aspect of exposure processing. It contains both credits and debits, and its size is equal to the sum of the credits minus the sum of the debits. Every exposure ultimately has one or more reserve lines.

A single exposure might have a single reserve line associated to it. It could also have several reserve lines associated to it if there are multiple "sets" of money for different purposes that must be tracked differently, for example, an exposure for a collision coverage could have separate reserve lines for auto body damage and glass damage because the carrier tracks these two types of payments differently.

A single reserve line might have a single check associated to it. It could also have several checks associated to it if:

- There are multiple payments (such as a recurring payment to treat ongoing medical care).
- There are multiple payees (such as a circumstance of litigation where the settlement amount is split between the claimant and the lawyer).

The relationship between policy and claim is one-to-one. Logically speaking, a policy can have many claims associated to it.

2.1.1 Claims Life-Cycle

A claim as a record of an incident has its own life cycle, image 2.3. From incident intake, claim opening, legal processes evaluation and its final payment there is different activities people and stages a claim has to go through.

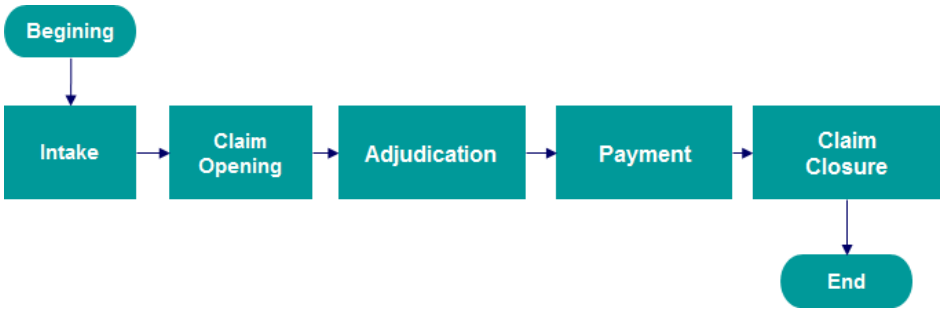

Figure 2.3: Claim Workflow

During intake, the carrier is informed of a potentially covered loss. For auto and property claims, this is often referred to as First Notice of Loss (FNOL). For workers compensation claims, this is also known as First Report of Injury (FROI).
The intake event is significant because:

- It is the point in time at which the claim is created and assigned to an adjuster;
- It is the point in time which determines events that must occur;
- It is the first major opportunity to control costs with regards to reducing leakage;
- It is the first major opportunity to reduce the time needed to process a claim;
- It is the first major opportunity to control costs with regards to business efficiency.

Adjudication is the process of determining if you are going to pay a claim, and if so, how much you are going to pay.
Within the insurance industry, adjudication is often further sub-divided into two phases: discovery/investigation and loss assessment.
During discovery (which is also often referred to as investigation), the carrier gathers whatever information is needed to process the claim. In some cases, all of the data may have been acquired during intake. In other cases, the carrier may need to do additional work, such as:

- Verify whether the loss is covered;
- Obtain statements from witnesses;
- Obtain information from third-party systems, such as police reports;
- Obtain estimates from the claimant or approved vendors.

During loss assessment, the carrier determines how much the claim is going to cost. This involves both payments made to claimants as well as expenses incurred in processing the claim.
This could also involve recovery, which involves situations where money can be acquired from other parties to help recoup the carrier's expenditures.
In general, assessments and estimates done by parties not working directly for the carrier fall into the realm of discovery/investigation. Assessments and estimates done by parties working directly for the carrier fall into the realm of loss assessment.
The discovery/investigation phase and loss assessment phase are not discrete. It is not unusual for a claim to bounce back and forth between these two phases as information is gathered and loss is assessed. Therefore, from the standpoint of the course, it is easier to address the two of them together at the adjudication level.

Once the loss has been assessed and the payment has been approved, the carrier can provide the payments to the claimants. The payment phase involves creating, approving, and issuing the checks.
One major concern of all insurance carriers is leakage/overpayment. Leakage is defined as "claims payments beyond those necessary to satisfy a typical claimant". Leakage could occur in many ways, including but not limited to:

- Paying for a loss that occurred at a time when the policy was not in effect, or for a loss which was not covered on the policy;
- Paying an amount in excess of the responsibility of the carrier;
- Paying an amount in excess of what the claimant needs to be indemnified.

In order to identify and control leakage, a carrier must closely track what payments are being made for. Consequently, the payment phase involves a somewhat rigorous categorization of claim costs (money paid to indemnify claimants) and expenses (expenses incurred as a result of processing the claim.

In some cases, the carrier may also attempt to recover funds paid to the claimant by obtaining money from a third party's carrier if the third party was at fault (which is known as "ubrogation") or by taking possession of a significantly damaged property and selling all or parts of it to a third party (which is known as "salvage"). Technically speaking, recovery does not need to start after

payment. Recovery can begin as early as first notice of loss if enough information is known at this time. For example, early in the claims process, the carrier could determine that the car is a total loss and immediately take possession of it and sell it to a salvage yard.

2.2 Financial Services Core Transformation

Insurance Companies and financial organizations are realizing that the do-nothing attitude towards technology modernization represents a risk and may actually cause a financial organization business to slip backwards as others keep pushing their services forward with new technologies [FORa].

Agencies spend almost half of the annual federal IT budget, $35.7 billion USD dollars, maintaining and supporting legacy applications [mou]. Legacy applications that don't provide today's customer needs.
Forty percent of customers say that they are losing trust in the industry and only 22% report they are gaining confidence [FORa].
Consumer loyalties to their financial organizations are on the decline and this could lead to irreparable attrition [FORa] - The business has to adapt.

Considering this, banking and insuring executives must take the lead to modernize its business - they need to build and pursue transformation. Four out of five financial organizations believe they will have to replace their core banking/insuring system in the next three to five years, with nearly 90% in favour of including SaaS or cloud based services as part of the infrastructure, according to a new survey of senior financial organization executives [Ban].

Replacing core systems in a financial organization is a very complex procedure, very expensive and organizations have to run both the new and the old systems in parallel for a long period to avoid shutting down the business [FORb].
The legacy systems were built 25 to 30 years ago and since then financial organizations have expanded their business to include wealth management, risk management, support for financial advisors, not to mention ATMs, online banking, mobile banking [FORb].

Many financial organizations planned to respond by working with new technology providers, even where these have fewer customer references, so that the financial organization can ensure that it implements standard service oriented architecture [Ban].

A slim majority of financial organizations still believe that IT is a unique selling point and will continue to rely on their own in-house development teams for niche technology that fits the financial organizations strategy [Ban].

2.2.1 TIA - The Insurance Application

TIA started in the heart of a Nordic Insurance Company with a single objective - make a core insurance platform that is portable across countries and companies, while rethinking the way that traditional insurance systems work [tia].

Core Solutions Transformation

It has evolved to allow migrations to the latest technology platform migrating from DOS based environment to client/server, to three tier technology, to a Service Oriented Architecture, and soon to an ADF version [tia].

In order to address client requirements, TIA is organized in 4 major assets:

- Policy: With dynamic policy validation; dynamic authorizations and referrals; clause triggers and product configuration accelerator;
- Claims: Issuing claims configuration accelerator; claims automatic excess calculation; claim sum insured validation reserve calculations;
- Account: Collection, UI Management and invoice details; payment method and frequency bulk update; exchange rate loading;
- Others: Print request screen; Report generation screen; Dynamic screen builder; Brokers access control.

For this four main assets there are eighteen modules, or project artifacts, to handle insurance business and workflow, table 2.1.

2.2.1.1 TIA - Claims Module

TIA Claims is focused on straightforward processing with minimum effort and optimizing the claims life cycle enabling you to increase your service excellence.
We can discriminate, as in the table 2.2, the key concepts in TIA Claims.

On figure 2.4 we can analyze how TIA handles a claim through out its life cycle.

- List of Claim Cases:
  - This form lists all claim cases entered in the claims system;
  - This form is used as a list-of-values for selecting an existing claim case.
- Claim Handling:
  - This is the main claims handling form where new claims are registered and existing claims are processed;
  - This form consists of several tabs and contains information related to both event and claim case level;
  - From this form is also possible access all other relevant claim handling screens for entering and maintaining information regarding the claim case, subcase, item, third party, extended relations, etc.
- Claim Case:
  - This form is used to enter and maintain details about a claim case;
  - Information already entered in the main Claim Handling form cannot be seen/maintained here. At the claim case level you can view and maintain such information as answers to questions (via the question class and question wizard), statistical codes, handler, informer contact, tasks, etc.

Table 2.1: TIA Modules Overview.

TIA Modules	Description
TIA Party	TIA Party records all types of parties in the same Name table. Each party is administered by role. An unlimited number of roles can be attached to each party. One or more relations can also be attached to a name
TIA Policy	TIA Policy allows direct, instant issuing of quotations or policies, or instant conversion of quotes into policies without having to repeat data entry. All necessary documents required for the customer are automatically produced, distributed, and stored in TIA's Archive or the insurers filing system
TIA Claims	TIA Claims handles the whole process of registering claims, gather additional information related to the claim, and managing correspondence, payment, follow-up, statistics and others
TIA Account	TIA Account is a comprehensive account payable and receivable module, allowing clients to have one or more accounts with the insurance company. Includes multiple currencies, payment module, flexible installments, reminders, full audit trail and general ledger
TIA Commission	TIA Commission keeps track of policies sold via internal, external agents or brokers. Commissions are calculated and transferred to the agent or broker account in the account system for later reconciliation
TIA Authorization	Functional authorization defines what kind of actions a user is allowed to perform within the TIA Solution. There is a number of screens used to administer and maintain user profiles and authorization
TIA Reinsurance	handles the process of registering reinsurance contracts, connecting contracts to programs, premiums and claim expenses, split premium/claim expenses between reinsurers and others
TIA Case Management	TIA Case Management's main functionality is registration, retention of documents, registration of warnings and tasks
TIA Marketing	TIA Marketing provides facilities for capturing all relevant information about a customer or prospect and makes it possible to view a customer's entire portfolio, set-up marketing plans/campaigns, campaign budgeting, follow-up and cost control
TIA Product Editor	The TIA Product Editor provides a tool to create and maintain products and their components. The Product Editor System consists of several forms where it is possible to define different product components needed to build a product
TIA Claims Editor	The TIA Claims Editor provides a tool to customize the way a claim should be manage. It allows to configure the claim history, questions, tasklists, default estimates and handlers for each claim type
TIA Complaint	Allows customers to keep track of the received complaints from reporting to closure
TIA Language	Maintains the configuration of different languages for labels messages and other contents
TIA Print	The Print module allows to set up and execute printing as well as maintain the print system data
TIA Batch schedule	Manages execution of programs in batch. In the Batch System it is possible to set up batch jobs and execute and control them
TIA Menu	The menu module enables a new user to navigate through pull down menus, using the mouse or the arrow keys, while the skilled user can branch across the menus using numbers for the desired screens
TIA Help	The Help module is based on context specific help and drill-down (zoom). It includes field help, technical help, functional use, etc.
TIA iCom	Provides functionality for handling communication with external endpoint environments

Table 2.2: TIA Claim Modules Overview.

Main Concepts	Description
Event Type	Event type that caused the accident.
Cause	Reason for the occurrence of the event.
Subcause	Detail of the reason for the occurrence of the event.
Third-parties	Entity that assumes the role of victim or guilty.
Involved Entities	Entity involved in the accident but not injured or considered as guilty.
Estimate	Represents the claim provision. Multiple estimates can be opened by claim. Each created estimate is associated to a certain cover and allows the creation of payments and / or payment plans.

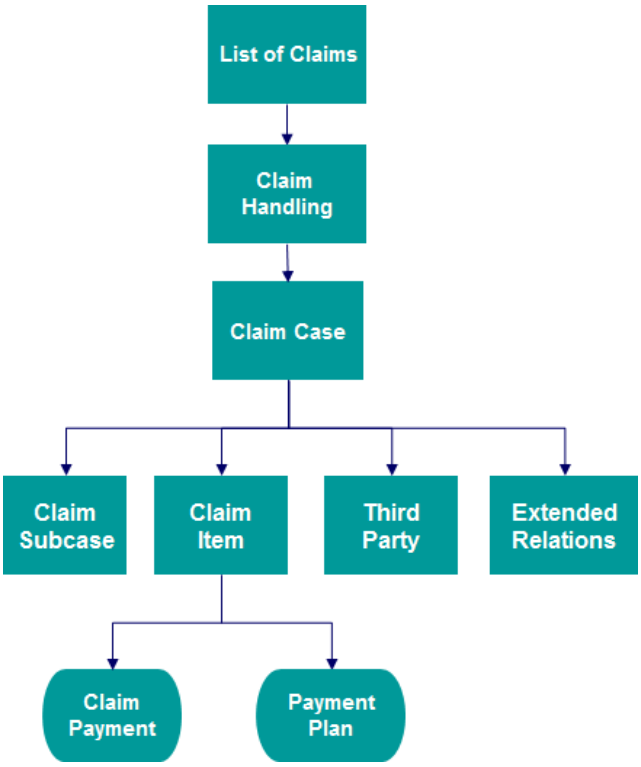

Figure 2.4: TIA - Claim Modules

- Extended Relations:
  - In this form is possible to view and maintain the relations between the party records in the NAME table (customers, companies, etc.) and other types of records in the TIA Solution (policies, claims, accounts, affinity groups, etc.);
  - It can, for example, be a list of all relations relevant to the claim case, such as Witness, Repairer, etc.
- Third Party:
  - This form maintains information about third parties involved in a claim case. The third party could be an accidental person, other party, etc;
  - Information like name and address of third party is kept in the general name entity. If this is not known, a free text can be used.
- Claim Subcase:
  - A subcase is related to one claim case or a third party;

  - A subcase contains data about the damaged items or a group of damaged items that are involved in the claim case. This could be a car, a building, a ship, some cargo, a person, or even an administration cost;
  - Different subcases within a claim case are normally defined when different parts of a claim case have different handling time;
  - In this form the handler register the main administrative information regarding the subcase e.g. whether the company covers this subcase or not, reference to object (covered object of a policy line), etc.

- Claim Item:
  - This form is used to enter and maintain estimates for claim cases and subcases. Reserves are created and posted on a claim once items have been created;
  - From this screen is also possible proceed to making claim payments based on the created items;
  - Claim item is also known as (Claim) Estimate, Payment Item or Claim Reserve.

- Claim Payment:
  - This form is normally used to make a single payment on a claim;
  - The form can be used for showing existing payments or making new ones. All payments are made against available estimates (specified on claim items);
  - When payments are made, new estimates are set that can be adjusted by the handler;
  - This form is pre-filled with the relevant data about one or more claim items if you have selected the check boxes next to the claim items to be included in the payment in claim item Form, before clicking the Payment button to access this form;

- Claim Payment Plan
  - This form is used for creating regular payments on a claim and splitting claim payments over a certain time period;
  - Using this form is possible to set up payments to be paid out to or collected from a certain party on a specified frequency basis.

2.2.2 Guidewire

Guidewire provides Core Systems and Data Management tools to insurance companies, helping them run their business in a more efficient way. Designed to be quickly customized for each customers line of business and needs, it can support numerous different types of Policies, Claims and Bills with just a few changes on the configuration files.

The InsuranceSuite is a set of four core products 2.3, by Guidewire, that meet the needs of insurance carriers, helping them expedite their policies, billing, and claim requirements.

Table 2.3: InsuraceSuite products overview.

	Product	Desciption
InsuranceSuite	PolicyCenter	Create, modify, renew cancel policies
	BillingCenter	Bill policy holders for policies and pay commissions for polucies to producers
	ClaimCenter	Process claims to provide payments to claimants when covered losses occur
	ContactManager	Store, Create and modify contacts.

2.2.2.1 Guidewire Architecture

All Guidewire applications are run within an application server, Java Enterprise Edition (Java EE) server. The platform was known as Java 2 Platform, Enterprise Edition or J2EE until the name was changed to Java EE in version 5.

Each Guidewire application is built and commonly deployed as an Web Application Archive (WAR) or Enterprise Application Archive (EAR) file to the application server. It contains all the configuration, operational data and data definition files necessary to execute the application. The application server is typically installed on a dedicated machine that does not host any other aspects of the architecture, such as the database.

The Guidewire application, at the center of image 2.5, consists of a set of business rules, a user interface, a set of integration APIs, and a data model. The technology for configuring these application elements is platform-level (and common to all Guidewire applications), but the specific rules, UI, APIs, and data model for each application are distinct.

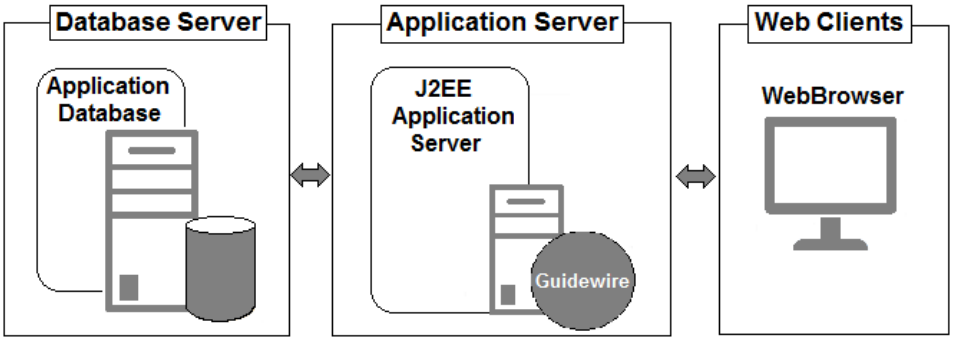

Figure 2.5: Guidewire Architecture

A relational database is a collection of data structures and objects related to each other. Databases store data in database tables. The operational data for the Guidewire application is stored inside a relational database. This database is typically hosted on a machine other than the machine hosting the Java EE application server.

The WAR file contains a configure file that specifies which database to connect to and how to establish the connection, this is useful when developing an enterprise application, switching between

machines depending on the project stage.

End-users connect to the application using a web browser. Each application generates a collection of standard HTML pages that are rendered by the browser. Each application is a server-side application. It dynamically generates the HTML pages and data within those pages. It makes only a minimal use of functionality in the client. Because there are no static HTML files, you cannot use the browser's Back button. All navigation is controlled on the server side.

In a production environment, configuration changes are typically made by rebuilding and redeploying the application's WAR file. This always involve stopping and restarting the application.

2.2.2.2 Guidewire Configuration

Three-tier architecture, figure 2.6, is a client-server architecture in which the functional process logic, data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is one of the industry-standard software architectures.

A data model entity is an abstract definition of a group of objects used by the data tier, such as Address Book contacts, or ABContacts, 2.6. It defines the information about the objects that must be stored in the database, such as Name, PublicID, and CreateTime. It is defined in a set of one or more XML files.

In most cases, each data model entity corresponds to a table in the database. The data model entity definition defines the table structure. Each instance of the data model entity is stored as one row in the database table.

A Gosu class is an abstract definition of a group of objects used by the application tier, such as Address Book contacts, or ABContacts. It defines the information about the objects that must be maintained in the application server's run-time environment, such as Name, PublicID, and CreateTime. It is defined in a set of one or more Gosu files that have a proprietary format and a ".gs" extension.

For every data model entity, the Guidewire application automatically creates an internal Gosu class with the same name. For every field in the data model entity, there is a field in the corresponding internal Gosu class. For example, the ABContact data model entity has a "Name" field, and the internal ABContact Gosu class also has a "Name" field.

Whenever the application needs to work with an instance of a data model entity (which is stored as a row in the corresponding database table), the application creates an instance of the corresponding Gosu class. The information from the database is then read into that instance. For example, if a user searches for the ABContact whose name is "Express Auto"" then the application finds the row in the database table for ABContact, creates an instance of the ABContact Gosu class, and reads the data from that row into that instance.

A PCF (Page Configuration File) is an abstract definition of a form or location used by the user interface tier, such as an "ABContactSummaryDV" detail view used to display information about

ABContacts. It defines the information about the objects to be displayed in the user interface, such as a Name field, a PublicID field, and a CreateTime field. It is defined in a set of one or more XML files using the proprietary PCF XML schema definition.

There is a strong correspondence between data model entities and internal Gosu classes. Every data model entity has one internal Gosu class. There isn't necessarily a strong correspondence between internal Gosu classes and PCFs. The data for one Gosu class could be displayed in a single PCF, or it could be displayed across multiple PCFs. The separation of the user interface tier and the application server tier gives developers the freedom to display data in whatever way makes sense to end users without being constrained by how that data is maintained in the application server or how the data is stored in the database.

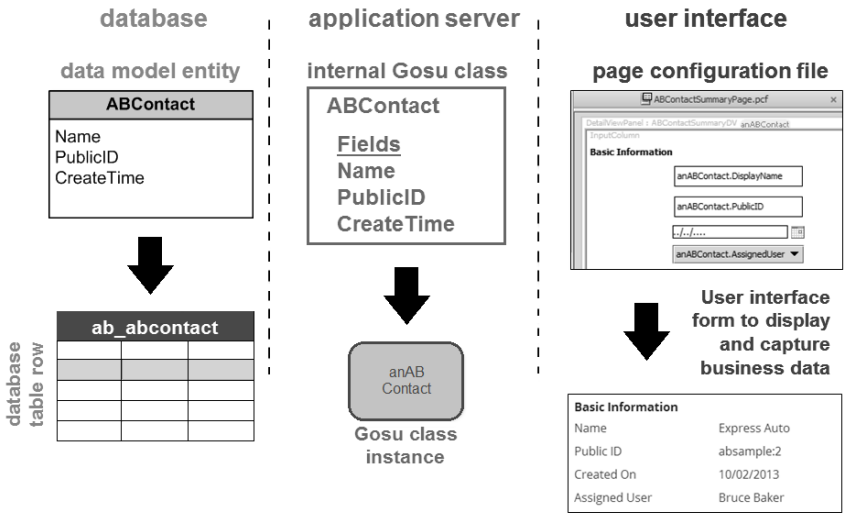

Figure 2.6: Guidewire configuration technology - Three tier

2.2.2.3 Guidewire Gosu

Gosu is a open-source publicly available programming language. Gosu has elements of both procedural and object-oriented programming languages and is similar to JavaScript and Java [Gui]. It runs under the sophisticated Java Virtual Machine (JVM).

It executes fundamental application behavior, manages complex business processes, executes hierarchical business rules and specifies dynamic user interface behavior, table 2.4.

It's with Gosu that the developing team configures and modifies Guidewire Suite.

2.2.2.4 Guidewire - ClaimCenter

In Guidewire ClaimCenter, the claim is created during the intake process by a CSR or adjuster through the new claim wizard. Some carriers may also use a portal application (such as Guidewire Claim Portal for Policyholders) to allow customers to submit claims themselves without Guidewire user intervention. The user (or policyholder) who creates the claim must specify the relevant policy and the claimants. They also typically detail as much information about the incidents as possible,

Table 2.4: Gosu actions on different working areas.

Working Area	Action
Business Rules	Apply specific programming logic for testing a condition and performing an action.
Entity Enhancements	Extend entity functionality with programming logic as entity methods
Entity Names	Programming logic that defines how to display a name for an entity instance
Gosu classes	Encapsulate data and code for a specific purpose or function
Workflows	Run custom business processes asynchronously, optionally with multiple states that transition over time

as well as any other loss detail information available at that time. Services may also be created on the claim or related to incidents.

Once the claim is created, ClaimCenter business rules segment the claim, which is the act of classifying the claim (for example as a “fast track”, “normal” or “complex” claim) so that the right strategies can be used to process the claim. The business rules also assign an owner to the claim and create a series of activities known as the workplan which identify work that must be done to process the claim. All of this is referred to as “claim set-up”.

Once the decision to pay on the claim has been made, exposures are created. Each exposure tracks a potential payment from a single coverage to a single claimant. Exposures can be created automatically by business rules or manually by adjusters.

A reserve line is an amount of money set aside for expected payments related to a given exposure. However, an exposure can have multiple reserve lines, which occurs if there will be two or more payments from the exposure and the carrier wants to track the money separately.
Reserve lines can be created automatically by business rules or manually by adjusters.

Most or all of the work to be done on the claim is detailed in the workplan activities. Once reserves have been created, those activities are completed.

The claim and each of its exposure have a maturity level 2.7, often referred to as a validation level. Typically, the final level is "ability to pay". When a claim is at ability to pay, checks can be written for it. When an exposure is at ability to pay, the money in its reserve line(s) can be used for checks. Gaining the ability to write checks against a claim and use the reserve lines of its exposures is a prerequisite to making payments.

Figure 2.7: Claim maturity levels on ClaimCenter.

In some cases, the claim and its exposures may become payable as a natural result of the completion of all of the activities. However, the two are not required to be functionally connected.

Furthermore, activities typically focus on tasks that must be done by a given time and/or must be done by people other than the adjuster. Therefore, it is not unusual for a claim to have all activities complete and yet the claim itself or one or more of its exposures is not payable. (It is also possible that a claim and all of its exposures are payable, but there are still open activities.) Once the claim and its exposures are payable, payments can be created. Money is transferred from the carrier to the payees through checks. Each check gets its money from one or more reserve lines.

When the financial obligation associated to an exposure has been satisfied, the exposure can be closed. If an exposure results in one payment, then the exposure is typically closed when that one payment is made. If an exposure results in multiple payments (which could occur if there is a recurring payment, possibly for ongoing medical treatment), then the exposure is typically closed when the final payment is made.

When all of the activities are complete, all the payments have been made, and all the exposures are closed, the claim can be closed. In some cases, a claim may be closed as soon as the payment is made (and the last indemnification exposure is closed). In other cases, the claim may remain open beyond the last payment (possibly because the claim involves recovery which is taking place after the last payment, or possibly because there is lingering activity work to complete, such as verification that legal documents have been filed with the appropriate government agency).

Documents and notes are typically used for every claim, but they typically record things that have occurred. They do not represent work that needs to be done, and in most cases their existence does not move the claim forward in the claim process. Consequently, they have been omitted from the discussion. Matters are relevant to moving a claim forward in the claims process, but only for claims involving potential litigation. Since these types of claims are not the most fundamental type of claims, matters have also been omitted from the discussion.

The business process steps and functional process steps do not have a one-to-one correspondence, image 2.8. In particular:

- During intake, the claim is created. Exposures and reserves could also be created at this time.
- During adjudication, ClaimCenter exposures could be created, reserves could be created, activities could be completed, and the claim and its exposures could become payable.
- At payment, payments are made. But, reserves could be created at this time, activities could be completed, and the claim and its exposures could become payable. Payment may also cause exposures and the claim to be closed.

2.2.3 Conclusion

Guidewire is a much more integrated platform, comprising all systems, such as databases, configurations, rules and validations on one system only - a developer do not have to go through different software to do the job. It is easier for developing, team collaboration and having a much wider view on the project and its evolution.
Guidewire makes its core objective to offer clients with high volume of data the best services, so it establish itself on a much wider market - North America.

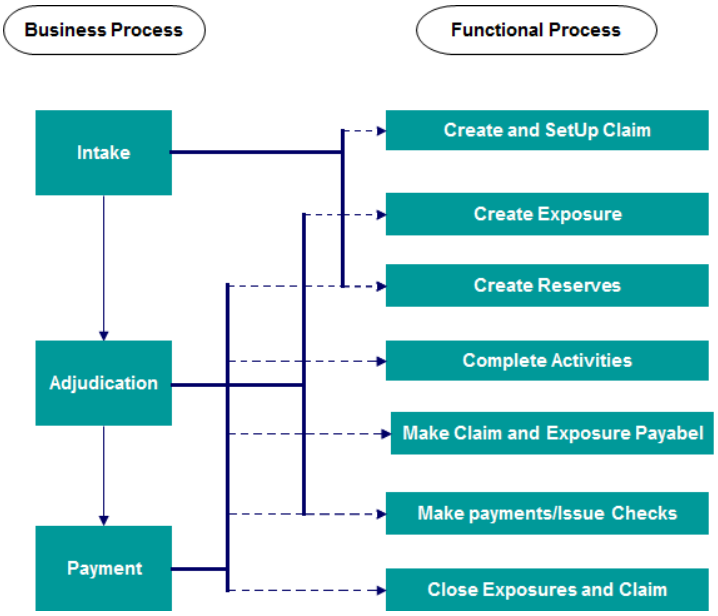

Figure 2.8: Business and functional claim processes

TIA on the other hand is a much more mature software, its development and evolution results from a continuous increment on it's Software and extensive requirements gathering directly from the business.

For such specific tools, and being clients so scarce, the company which marketeers it's product the best wins, and in terms of geographic positions there is a clear relation between companies on Europe choosing over TIA and overseas in North America, insurance companies tend to choose Guidewire - not so much of a statistical fact that i can point out to, but an idea that resulted from a conversation with a Guidewire specialist itself.

Chapter 3

Project Management

The project have taken place at Canada - Toronto, for one insurance company. The main objective of this project is to completely transform it's core solution. The client is currently working with a system with more than 20 years. It can no more keep up with the insurance company needs and ambitions.

The project was landed by Deloitte Canada whose partned with Guidewire as main technical provider. Deloitte Portugal arrived later, building a strong team of professional FSI-IT developers.

This chapter will provide some insights on the project the student was allocated to. A brief presentation of the different entities involved on the project, work methodologies, releases and increment strategies will be made.

3.1 Project Releases and Increments

For this major core transformation project, some main delivery strategies were taken. The project was subdivided in three major releases: Realize 1 (R1) contacts will be migrated from legacy system to the contact module on Guidewire (Contact Manger), new contacts, from this release on, will be stored and processed under a server running this module;

Release 2 (R2) will then be delivered. Policy Center will migrate all policies from legacy system to a new machine running this module and new policies will from now on be stored and processed on this instance. Billing Center module will also be developed and delivered under this release. Billing Center will support all billing processes of the company;

Release 3 (R3), release in which the student have worked with. Claims will be migrated from legacy system, new claims and records will be stored under Claim Center machine, payment calculations and adjusts will from now on be processed with Guidewire, manual intake processes and automatic intake processes will store it's information directly in this machine.

In this last release duo to its extensive development increment strategies were implemented: Form the first increment, Increment 1 (IRA), claims intake, registration and adjudication will be delivered. At increment 2 (CM) Claims management will be developed and delivered. For the last increment, Increment 3 (Benefits) all claim's benefit processes, calculations and adjusts are to be implemented and delivered.

It was at this very last increment of release 3 that the student was allocated to. The challenge was to understand the benefits workflow, business logic and develop well build solutions for the tasks assigned to him.

3.2 Agile

Agile facilitates the interaction between all the project entities. There is a daily hour were people get together in front of the agile board and discuss it's state tasks and future strategies.

The entities involved on agile on the project the student has been allocated are:

- Business: providing some updates on internal meetings and what is expected with certain card;
- Developers: there is an update on the development state and outputs obtained from the solution, issues and time extension are also discussed;
- BA: requirement analyses, update, project strategies and team management.

The agile board is composed in six stages, 3.1:

- Requirements: requirement sessions take place in this stage, discussion of objectives and deadlines are established. There is a strong business and BA presence at these meetings - there is also a light presence of developers at the sessions;
- Design: Continuous discussing between developers and BAs, requirements are analyzed and some clarifications with the business may be necessary;
- Ready for build: A queue of cards waiting for development;
- Build: Sessions between developers take place. Tasks for each member of the team are established. Development of the technical solution;
- Test: Sessions between developers/BA and BA/Testers take place. Scripts of tests are established. Communication between developers and testers are frequent;
- Ready for Demo: A queue of cards waiting for appointments with business. This sessions take place at the Client's office. For this sessions the developer is the one who leads the presentation and demos the solution. Feedback from business is also frequent. BAs are also strongly present at the sessions as there is a lot of requirement changes;
- Accepted: Final state of the card, if successfully accepted from business.

At any given time this steps can iterate going forward or backwards, as figure 3.1 represents.

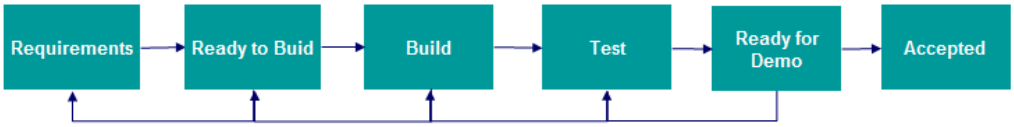

Figure 3.1: Agile organization at the project.

3.3 Conclusion

The project context, geographic location, different language factor and technical challenges made this project a good learning source, professional character development and work ethic and rhythm improvement.

Project organization by releases seems to ensure that delivery dates and consistency on the quality outputs are met. For big releases, increment strategy facilitates team organization and objective definition.

Agile proved to be a solid foundation for project development. Its methodology: flexible and really communication focused; improves working speed avoids erroneous development and improves consistency on the desired output.

The student have been related to the increment 2 of release 3 - allocated to a team specialized in payments processes. He focused on payment sequences: the distribution and definition of a claimant's earnings for a period of time and the technical problems it evolves.

Chapter 4

Task 1 - Overpayment Write-Offs

4.1 Overpayment Overview

For any benefit paid to the worker which exceeds the amount a person is entitled for, an overpayment is created. This overpayment can be caused by multiple factors, to name a few: the insurance company paid benefits for a time period for which the worker has also been receiving earning from the employer; the worker has provided false or misleading statements to the insurance company in an attempt to obtain unlawfully benefits; duplicate submissions of the same claim resulting in a duplicate payments.

It is important that this payment is promptly returned to the insurance company[ove05], in most cases the payee itself returns the over payed amount but if not actions are taken to recover that amount. The most common method is to recover the overpayment by deduction of future worker benefits.

A Write-Off, as in this context, is the reduction of the overpayment value. The main objective of a Write-Off over a created overpayment is to allow the business to articulate manual corrections on the amount overpaid.

4.2 Task Objectives

The main objective of this task, was to add a logic of adding or subtracting an amount to the overpayment.

This overpayment is associated with a worker, across all claims registered in it's name.

- A new overpayment status, No Further Action (NFA), is to be create:
  - a user won't be able to choose such status;
  - this status will be default for any new overpayment created;
  - any new downward adjustment on the overpayment will set it as NFA;
  - if completely inactivating a payment, the overpayment will default to NFA;
- Status, upon conversion from Legacy, are to be maintained (including NFA);
- All adjustments and changes to overpayments, debt causes, status changes and amounts (including how overpayment amounts are allocated), will create an entry to account payments in other financial transactions;
- Write-off amounts exceeding authority limits will generate an activity to the user's manager;

Table 4.1: Overpayment status conditions.

Condition	Status
If outstanding amount equals to NFA amount	NFA
If overpayment amount equals to total write-Off amount	Write-Off
If overpayment amount equals Recoverable amount	Recoverable
If pending amount is zero, recoverable amount is greater than zero	Recoverable
If NFA amount and write-off amount equals to full amount of overpayment	NFA
If any combination of write-off, NFA and recoverable amount exists, and recoverable amount is greater or equal to zero	Recoverable
If any combination of write-off, NFA, recoverable and/or pending amount exists, and pending amount is greater than zero	Pending
If write-off amount and NFA amount greater than zero but pending equals to zero and recoverable equals to zero	NFA

- The manager needs the ability to approve or deny the write-off:
  - if write-off denied, amount status needs to revert back to it's previous;
  - a notification is to be created to the case owner;
  - case owner are to be able to write-off/or make recoverable part or all overpayment amount and change status and amount allocation repeatedly as required;
  - if a write-off has been approved, case owner will still be able to return the amount to recoverable after the write-off, if required.

4.3 Solution Developed

Covering the overpayment status change requirements, the solution achieved can be presented as in the table 4.1. These conditions are defined in Guidewire rules so the overpayment status is updated whenever there is an change on the overpayment entity.

For the logic on the overpayments amount write-off, three core procedures were created:

- Write-Off Overpayment Amounts, algorithm 1: the entry point for the write-off logic;
- Create Net Recovery, algorithm 2: method that calculates a net amount across the overpayment and desired write-off;
- Authority Limits, algorithm 3: validating the user's authority limits against the write-off intended.

4.4 Conclusion

There was a new concept of net recovery created by the developing team. This enabled the logic on overpayments to handle write-offs, without saving its actual value - guaranteeing the consistency of the value. This write-off is then capable of, respecting all the client rules, to increase or decrease the total amount of an overpayment of the worker.

Algorithm 1 Write-Off Overpayment Amounts

procedure `writeoffOverpaymentAmounts`(*overpayment*, *WriteOffAmnt*)

```
1: if Overpayment is NULL then
2:     return
3: end if
4: if WriteOffAmnt is NULL then
5:     Display Write-Off Error
6: end if
7: if WriteOffAmnt > overpayment then
8:     Display Pending Amount Error
9: end if
10: NewWriteOff ← createNetRecovery()
11: if WriteOffAmnt is NULL then
12:     Create new overpayment with previous state {to ensure we keep record of writeoffs}
13: end if
```

end procedure

Algorithm 2 Create Net Recovery

procedure `createNetRecovery`(*overpayment*, *NewWriteOff*)

```
1: prevWriteOffAmount ← overpayment.totalWriteOffAmountAccrossClaims
2: adjustmentToWriteOffAmount ← NewWriteOff - prevWriteOffAmount
3: if adjustmentToWriteOffAmount ≠ 0 then
4:     recoveryWriteOff ← adjustmentToWriteOffAmount
5: end if
   return recoveryWriteOff
```

end procedure

Algorithm 3 Authority Limits

procedure `doesRecoveryNeedApprovalByUser`(*overpayment*, *user*)

```
1: AuthorityLimit ← getWriteOffLimit(user)
2: totalAmount ← overpayment.totalWriteOffAmountAccrossClaims
3: if totalAmount ≥ AuthorityLimit then
4:     return TRUE
5: else
6:     return FALSE
7: end if
```

end procedure

There was also some minor user interaction changes, allowing the user to, on the overpayment page, input the desired amount to write-off - repeatedly as desired.

As one of the first tasks on the project, this was more of a technical insights learning and getting up with the business quest - ended up doing some minor changes under supervision and adapting on the project working methodology.

Chapter 5

Task 2 - Capitalizations

5.1 Capitalization Overview

As in this project context, a capitalization is perceived as the amount yet to be received by the worker from the Insurance Company, across all his active benefits and benefit types. Not only a capitalization expresses the amount the Company has yet to pay the worker right away, it expresses an estimated value the company has to allocate for future expenses with this worker. In this sense there are multiple factors the company takes into account so a correct capitalization value is calculated, e.g: the worker age, gender and previous incidents.

5.2 Task Objectives

The main procedure of a capitalization can be summarized as follows:

1. Find the worker's age as of the current cycle date;
2. Use the appropriate look up table to find Capitalization Factor - depending on the benefit type paid, % of benefit, the current age and gender;
3. Multiply the Capitalization factor by the current monthly rate to come up with the capitalization value.

$$Capitalization = CapitalizationFactor * CurrentMonthlyRate$$

- The capitalization factor is based on age and gender on cycle date;
- Current Monthly Rate, is the total earnings the claimant is received per month;

But it was required by the business that Capitalizations were to be calculated differently for the three main benefits available, 5.1:

- Wageloss;
- NEL;
- Survivors.

If changes are made to any of the following: age, gender, monthly benefit rate, and an adjustment is made that increases or decreases benefit for a benefit type in which a capitalization has been applied, then:

- The final value of a capitalization is shown on capitalization's screen as debit, positive format;
- Capitalization screen shows one line for a credit (negative amount) of the previous capitalization amount for that benefit type. Shows a second line with a new debit (positive amount) for the new capitalization value;

Table 5.1: Capitalization specifications across different benefit types.

Exposure	Fields of Interest	Capitalization
Wageloss	• Cycle Date • Age • Gender	$MonthlyRate * Factor$
NEL	• Cycle Date • Age • Gender • Incident Date • $CycleDate - IncidentDate$ • Extra Months	$MonthlyRate * 1000/Factor$
Survivors	• Cycle Date • Age • Gender	$C = CurrentMonthlyRate$ $P = PreviousMonthlyRate$ $Cap = (C - P) * Factor$

Table 5.2: Capitalization Table - line example.

Capitalization Table	Benefit Type	Age	Gender	Rate
table name	e.g: Pension	65	M	value
...	...	...	...	...

- If a benefit sequence is inactivated, then Capitalization screen should show one line for a credit (negative amount) of the previous capitalization amount for that benefit type. The net total for that benefit type is $0;
- if a benefit is reactivated, then Capitalization screen shows one line for a debit (positive amount) of the new capitalization calculated using the capitalization formula.

If there is a future dated monthly benefit to be paid, then determine the type of table to be used based on the benefit type paid.
The Consolidated Capitalization Tables won't be shown in this work but basically they are based on Devitt's law. This law is based in several factors such as:

- accident date;
- indexing factor;
- gender;
- age;
- duration of the benefit;
- type of benefit.

The final table will result in something like the table 5.2.

Core Solutions Transformation

When decreasing or canceling a pension or Bill 165 award, figure 2.2, the capitalization table used to create the credit should be the same table that was used to create the original debit. However, because the sequences will break around other benefits, the system will generate two credits:

- one credit calculated on a short term table (for the broken sequences);
- one credit calculated on the life table (for the open ended sequence).

If Adjustment made that reduces the NEL benefit amount or Adjustment made that increases NEL benefit amount, then Capitalization screen shows one line for a credit (negative amount) of the previous cap amount for that benefit type. Shows a second line with a new debit (positive amount) for the new cap value.

If NEL award is adjusted but NEL percentage remains the same (Date of Birth is changed), then Capitalization screen shows one line for a credit (negative amount) of the previous cap amount for that benefit type. Shows a second line with a new debit (positive amount) for the new cap value.

When you inactivate a commuted NEL benefit, a negative cap value is not created. Currently NEL credits to the employer are manually processed.

5.3 Solution Developed

As the requirements were complex, so the solution developed. It all started by handling the three main benefit types of capitalizations:

- WageLoss;
- NEL;
- Survivors.

There is a main method being called: *CapitalizationCreatorParticipant*. This class overrides the function *runExecutionParticipant*, so it's automatically called whenever a wizard, in this case a payment wizard, is finished. On figure 5.1 we can have a general overview of the class *CapitalizationCreatorParticipan*, where the participant is located, and classes related to it.

One important aspect of this task is that we can split the benefit types into two major groups: a group where the capitalization is calculated based on the sequences of payments (WageLoss and NEL) and a group, in this case a group of one element (Survivors). On figure 5.2, we can see the two groups, its workflow and how each is being handled.

Algorithm 4 WageLoss get Capitalization Factor

procedure `getCapFactor()`
1: Age ← Claimant's DOB
2: Gender ← Claimant's Gender
3: Factor ← Factor On CapFactor Table //according with age and gender
end procedure

Algorithm 5 WageLoss Adjust Capitalization

procedure `WageLoss Adjust Capitalization`
1: Overlayed Sequence ← First Sequence That Overlaps Current Cycle Date
2: Previous Monthly Rate ← Overlayed Sequence's Monthly Benefit Amount
3: Current Monthly Rate ← Current Sequence's Monthly Benefit Amount
4: Factor = getCapFactor()
5: Cap ← (Current Monthly Rate - Previous Monthly Rate) * Factor
end procedure

Algorithm 6 WageLoss Create Capitalization

procedure `getCapFactor()`
1: Monthly Rate ← Sequences's Benefit's Monthly Rate
2: Factor = getCapFactor()
3: Cap ← Current Monthly Rate * Factor
end procedure

Algorithm 7 NEL get Capitalization Factor

procedure `Get NEL Cap Factor`
1: Gender ← Claimant's Gender
2: Incident Date ← NEL Benefit's Incident Date
3: DiffYears ← Years Between Cycle Date and Incident Date
4: Age ← Claimant's Age At Incident Date + DiffYears
5: Extra Months ← Months Between Incident and Cycle Date - DiffYears * 12
6: Factor 1 ← Factor On CapFactor Table //according with age and gender
7: Factor 2 ← Factor On CapFactor Table //according with age + 1 year and gender
8: $CorrectedFactor = (12 - ExtraMonths) * Factor1 + (ExtraMonths * Factor2)/12$
end procedure

Algorithm 8 NEL Adjust Capitalization

procedure `NEL Adjust Capitalization`
1: Overlayed Sequence ← First Sequence That Overlaps Current Cycle Date
2: Previous Monthly Rate ← Overlayed Sequence's Monthly Benefit Amount
3: Current Monthly Rate ← Current Sequence's Monthly Benefit Amount
4: Factor = getCapFactor()
5: Cap ← (Current Monthly Rate - Previous Monthly Rate) * 1000 / Factor
end procedure

Algorithm 9 NEL Create Capitalization

procedure `NEL Create Capitalization)`
1: Monthly Rate ← Sequences's Benefit's Monthly Rate
2: Factor = getCapFactor()
3: Cap ← Current Monthly Rate * 1000 / Factor
end procedure

Algorithm 10 Survivor Create Capitalization

procedure `Survivor Create Capitalization`
1: Beneficiary ← Sequence's Beneficiary
2: Current Monthly Rate = Get Monthly Rate For Current Cycle Date
3: Factor = getCapFactor()
4: Cap ← Current Monthly Rate * Factor
end procedure

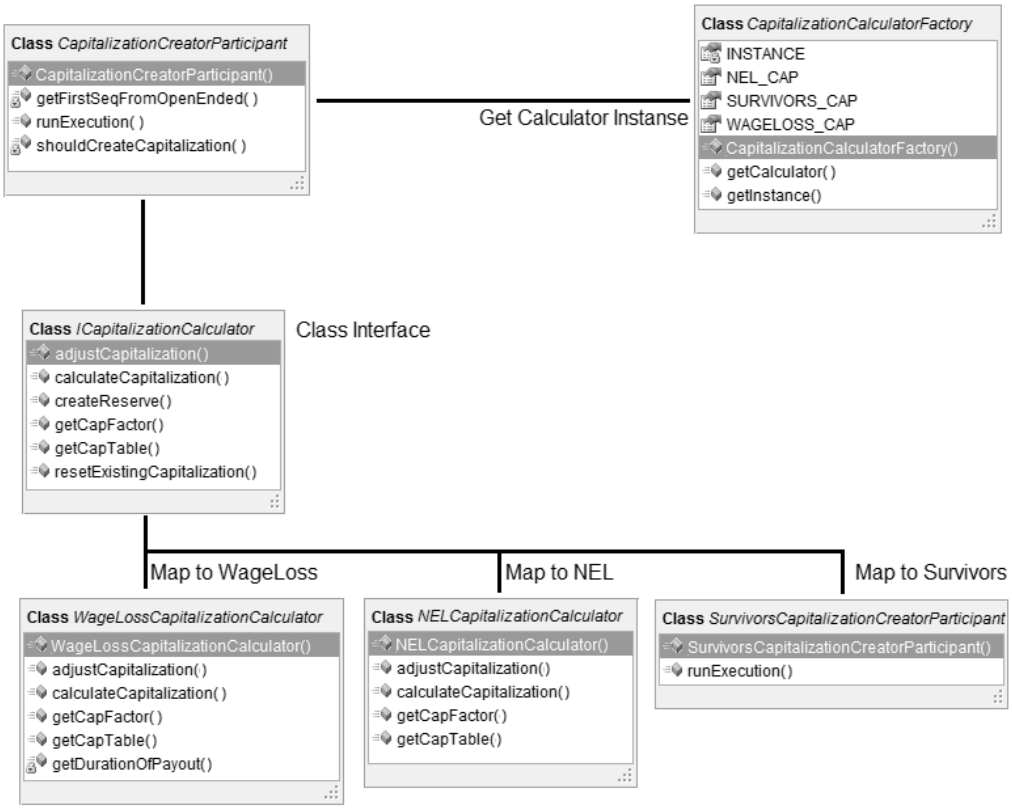

Figure 5.1: Capitalization Class Overview

5.4 Conclusion

The main challenges on the presented task were related to the extensive and complex business model. The requirement gathering was a difficult process. A lot of clarifications were needed and continuous requirement gathering sessions have taken place.

It was a sensitive topic for business to disclosure. Some information regarding this task was access restricted and there wasn't a central person whom had the general view of the topic. Communication with previous teams working on this task was also needed.
It taked a couple of Demos so this task could be correctly completed and accepted because there was always something more that needed to be done or not working as expected.

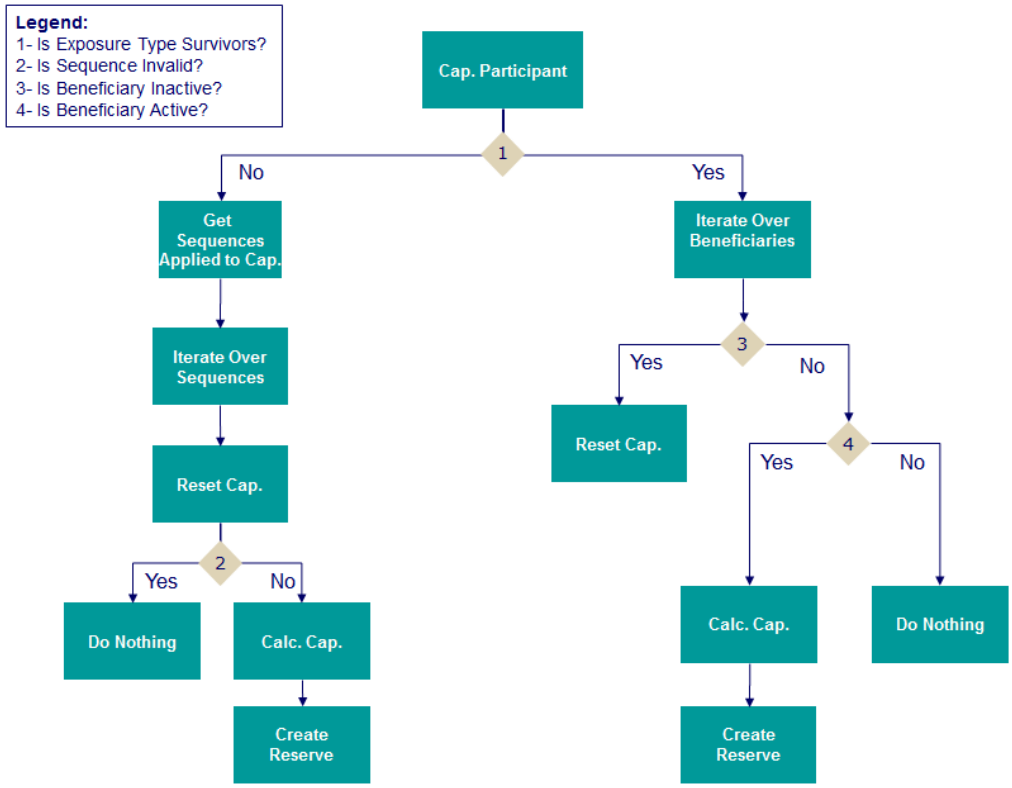

Figure 5.2: Capitalization workflow overview.

Chapter 6

Task 3 - Manual Check Overlimit

6.1 Manual Check Overlimit Overview

One possible way for the business to generate payments, is to manually create a check. As there are some authority limits, which defines the highest amount an Insurance professional can create, if he/she needs to make such over-limit payment, there are some hierarchical approvals this request has to go through.

In business logic, there will be a Payment specialist giving an order for a check to be created. The amount of the check will be compared with the maximum amount this specialist can create. If the check's amount is higher than the payment specialist limit, this check is required to have two approvals from the payment specialist's supervisors. The approval pattern is pretty straight forward: if we have an over-limit check there will be an activity assigned to the payment specialist's supervisor. This supervisor can either approve or decline the check. If decided to approve, the activity will be marked as first level approved and the person who approved it is required to assign this activity to another peer. When assigned to the second decision maker, if the activity is declined, the check will also be declined, if approved, this activity will go to a treasury queue where the payments are processed.

6.2 Task Objectives

Review Overlimit Payment activity to be available and created manually. A check amount field and flag (to determine whether its NEL or not) will be available upon the creation of the activity.]]
The created activity will store a link to Legacy System screen. It becomes active once the second level approval approves the activity. Based on the flag, the activity is assigned to NEL Clinical Specialist's manager.
Once the activity is created, the delegation of activity is manual, second approval will be kept as a record and upon final confirmation of the last approver, the activity is assigned to the Treasury Queue.

This treasury queue is maintained and used as reference for payments of manual checks created. If a check amount is higher than the user limit for such action, a Overlimit memo will be auto-attached to the activity, but not available on the claim file.

In the activity details add a visible field to present total amount of the check (manually entered) and flag (NEL or not).
If the Overlimit is denied the activity is sent/returned to the person who created. It will store a trail of the person denying it, being this the first or second approver.

6.3 Solution Developed

The logic of the new payment procedure can be translated as in figure 6.1.
A new activity was created by the student, allowing two levels of approval as required by business, auto-assign the activity, when creating an Overlimit payment, to user's supervisor, allow supervisors to approve and assign the activity to its peers and as it gets the two levels of approval required, this activity will be assign to the treasury queue. This is a queue for payments to be made and processed by the treasury staff.

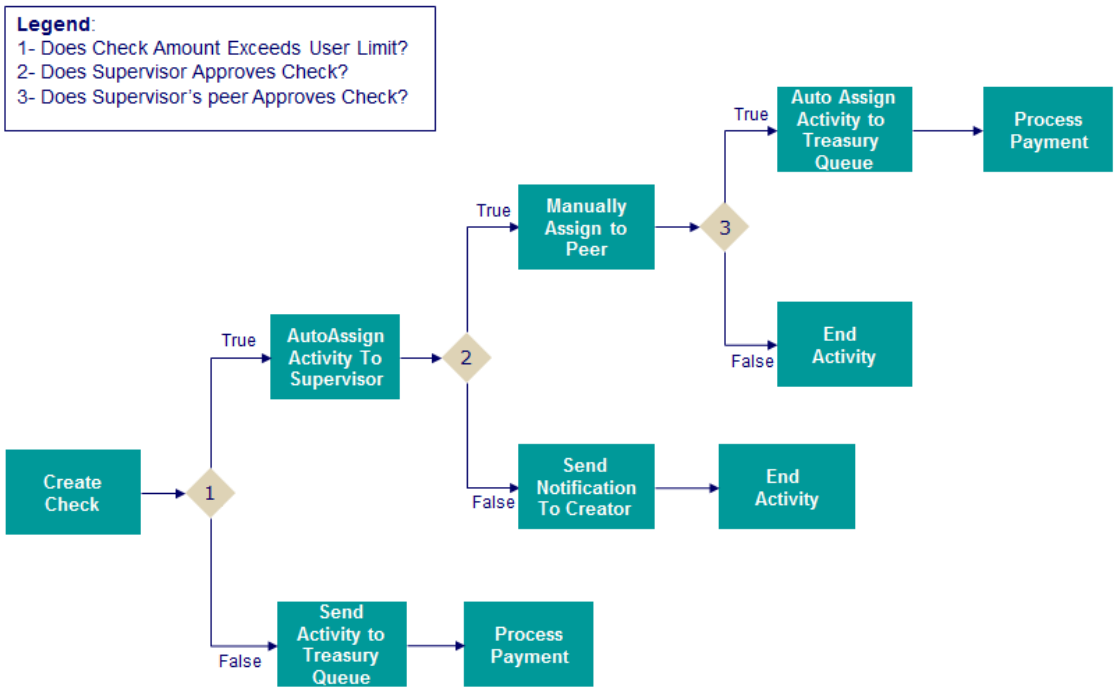

Figure 6.1: Manual check overlimit workflow.

6.4 Conclusion

Manual Check Overlimit was the first task where the student was set as main responsible for the development and delivery of the card. He attended the requirement sessions and made his pursuit for clarifications needed so he could build the solution with the expected output. He was also the one who conducted the demos.

Chapter 7

Task 4 - HATS - Legacy Integration

7.1 Legacy Integration Overview

As one of Increment 2 strategies, it was required by the business to integrate the system in use by company on the new project.
The reason for this is that all benefit's payment processes and logic were left out by project's release 3, increment 3. A safe bet, was to embed the old system on the new one and allow users to experience a one solution work-flow in the current project's release.
All claim's intake, contacts storage and processes will be taken into account inside Guidewire, although all payments will be processed in Legacy.
There is a migration sub process that needs to be implement because only new claim records will be fully stored and processed on Guidewire. It's the ones yet to be migrated that will present the links to the old directory on legacy system containing the claim record.

We can discriminate the following values on this task:

- Customer Experience

Key stakeholders, by having direct and integrated access, will experience optimal interactions and benefit from increased customer services by case owners and others. The ability to have a global view of benefits, payments and transactions will enhance and speed-up the process. This process will also reduce potential mistakes by ensuring that the screens are synchronized between Legacy and ACES.

- Enrich User Experience

ACES was built for full screen size. Not being able to maximize the screen will limit it's user experience and productivity. Legacy and ACES screens cannot be opened in separate windows and seen side-by-side without major impacts. Major costs would be added on the project and extensive requirements gathering would be necessary.

- Health and Safety

Toggling between screens of ACES and Legacy with a minimum of 200 times per day (averaging from a case load of 25 transactions x 8 hours of work) is a major health and safety issue and is already prevalent in ACES staff. Health and Safety must be built into the design - embedding the whole project in just one solution.

7.2 Task Objectives

For this task, business wanted to use legacy payment module inside the web context of Guidewire, figure 7.1.

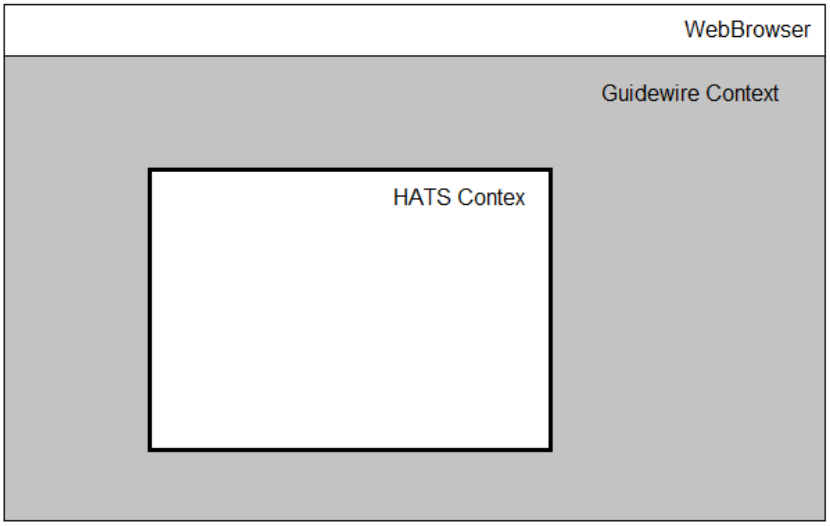

Figure 7.1: Web solution display.

Auto Log On - No separate log on for Guidewire web platform and Legacy system, was an objective for this task. Claim Sync with Legacy records, navigation to desired page, controlled by attributes passed from Guidewire.

When user selects a link to Legacy, the active claim number within Guidewire will point out for the same record on Legacy storing system. The User should be able to update his/her legacy password in Guidewire ClaimCenter. It is required to keep Legacy screens sessions active when navigate away and back within Guidewire web application. Users can continue their transaction from where they left off.
The user is able to navigate through the legacy screens, by manually interacting with legacy. Legacy screens will be embedded within Guidewire browser window. User will have the ability to jump to legacy screens using information passed from the URL, it will trigger automatic processes and navigate to the desired screen.

The solution should be able to maintain it's current session, on Legacy system side, during a fixed period of time.

Different Legacy links inside the web app jumps to different screens in Legacy.

7.3 Solution Developed

For the two technologies to work together (IBM Legacy and Guidewire) a bridge to connect both is required. The solution found was HATS - Host Access Transformation Services. This software, developed by IBM itself, can create Web applications, including portlets, and rich client applica-

tions that provide an easy-to-use graphical user interface[HAT].

Some links inside guidewire web app, are pointing to HATS solutions. This trigger a series of Macros and screens customization, on HATS side, with the final output being a screen dump of the actual state of the Legacy Screen System.
Whenever a user clicks on a link, an array of parameters are passed on the URL, pointing to Legacy System Machine.
Security issues are to be handled. The URL, is encrypted and encrypted, on Guidewire and HATS respectively.

The actual architecture for this Task can be presented as in figure 7.2.

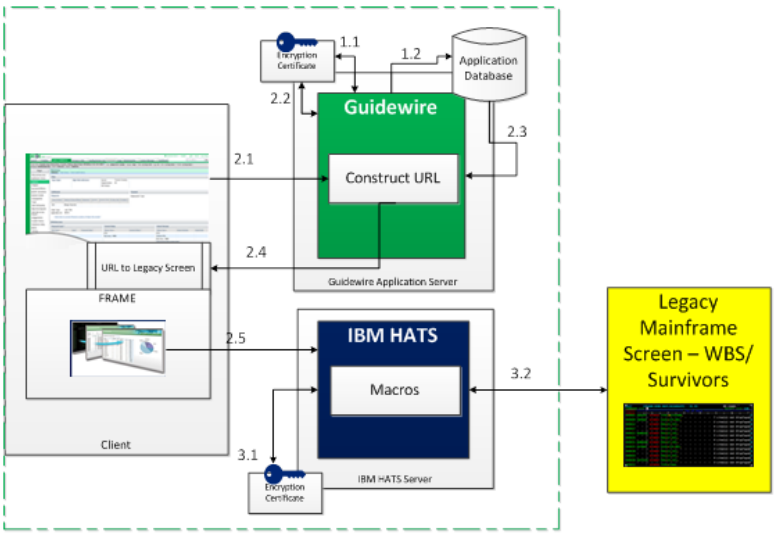

Figure 7.2: HATS - Solution Architecture.

The solution will then be deployed into a Websphere Application Server (WAS) - which hots java based applications. From WAS we will both: by telnet communicate with the old legacy server and over HTTP communicate with Web Browser. Figure 7.3, for reference.

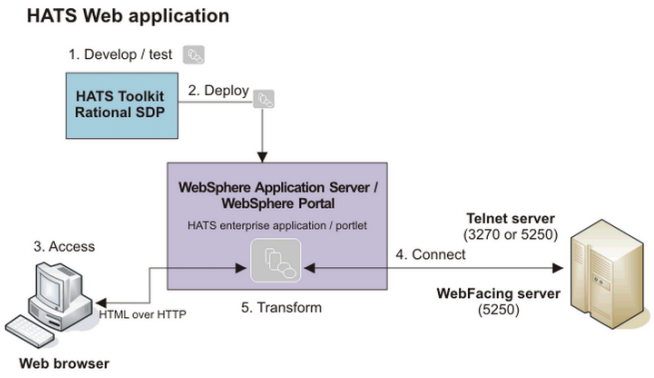

Figure 7.3: HATS - Solution Deployment

7.4 Conclusion

The Developed solution, allows the user to have a single application experience.
It will grant the access to legacy system by clicking in simple links and tabs on the actual web context of the Guidewire Server Application.
This link is set to perform automatic steps inside the actual legacy system. Therefore the user will experience some lag when trying to do this action.

On Guidewire application side, the user is expected to store it's credentials by inserting them in the administration tab of the web solution.
This information, along other parameters are being encrypted and decrypted when data is transmitted on both sides of the communication.
Based on this parameters information, some Macros are set to return a screen dump for the desired screen that the user wants to navigate to. Navigation was also a challenge for this task, as it required the team to recognize and handle the intermediate steps and screens between both ends.

Chapter 8

Conclusion

Two main challenges were found while on this project:

- Technical Challenges

Working with such a massive tool like Guidewire proved to be a difficult task. Even for those familiar with the programming language and configuration techniques. Solutions must be developed taking into account the whole project context. Impacts have to be analyzed and new extensions must be integrated with what is already done.

It is also a challenge for the developer to communicate gracefully with those who might touch the code later. Solutions are not always perfect and there is a need to articulate the development with different people. Good coding practices are needed for better development performance. Fixes and gaps can be easily developed reducing risks.

- Business Challenges

The Insurance Industry is not a simple business to understand. It has a lot of rules, concepts and exceptions - it is the result of years and years of constant adapt and evolution.
Achieving the functional knowledge and being able to envision the project as a whole, will always improve the solution. Errors and information conflicts are reduced as problems are trimmed down to basic achievable steps.

Working on a different geographic location was a big adventure - the best I can take from it is the improved English I developed, as in writing and speaking.

My critical thinking also improved a lot - it had to. The gap between what the business means and how it can be technically executed is huge, problem solving skills are

It was also interesting to see how a project this size works, there is no magic, just people with different characteristics and talents working together to make a final product. At the end a structured and organized team will prevail.

Bibliography

[Ban] Eighty % of banks to replace core systems "within five years" [online]. Available from: http://www.bankingtech.com/284751/80-of-banks-to-replace-core-systems-within-five-years/ [cited 17 August 2015]. 8

[dela] About deloitte [online]. Available from: http://www2.deloitte.com/us/en/pages/about-deloitte/articles/about-deloitte.html [cited 10 August 2015].

[Delb] Big four - deloitte [online]. Available from: https://en.wikipedia.org/wiki/Deloitte [cited 15 September 2015].

[delc] Deloitte consulting llp at a glance [online]. Available from: http://www.vault.com/company-profiles/management-strategy/deloitte-consulting-llp/company-overview.aspx [cited 11 August 2015].

[deld] Deloitte profile on forbes [online]. Available from: http://www.forbes.com/companies/deloitte-touche-tohmatsu/ [cited 11 August 2015].

[FORa] [online]Available from: http://www.forbes.com/sites/oracle/2013/08/14/beat-the-core-transformation-conundrum-by-acting-boldly/. 8

[FORb] Core banking replacement remains locked in the future [online]. Available from: http://www.forbes.com/sites/tomgroenfeldt/2013/09/10/core-banking-replacement-remains-locked-in-the-future/ [cited 17 August 2015]. 8

[Gui] Why gosu? [online]. Available from: http://devblog.guidewire.com/2010/11/11/why-gosu/ [cited 30 August 2015]. 15

[HAT] What is hats [online]. Available from: http://www-01.ibm.com/support/knowledgecenter/SSXKAY_8.0.0/com.ibm.hats.doc/doc/gsintro.htm [cited 06 August 2015]. 37

[mou] The danger of legacy systems [online]. Available from: http://web.archive.org/web/20120323165836/http://www.mousesecurity.com/?p=220 [cited 17 August 2015]. 8

[ove05] Benefit-related debts/overpayments [online]. 2005. Available from: http://www.owa.gov.on.ca/en/appeals/Pages/Benefit-Related-Debts-Overpayments.aspx [cited 08 July 2015]. 23

[tia] The tia history [online]. Available from: http://www.tiatechnology.com/en/insurance-solutions/tia/history/ [cited 14 August 2015]. 8, 9

Glossary

Beneficiaries	People that are dependent on the claimant.
Legacy	Old, obsolete or out dated system, application, method or technology.
Liability	Legal debts or obligations
Litigation	An action brought in court to enforce a particular right
Lump-Sum	One-time payment for the total or partial value of a benefit.
Portlets	Fragments of Markup Language managed and aggregated into a portal.
Write-Off	A reduction in the value of an asset or earnings