TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Tatjana Kruglova 142442IAPB

MITIGATING RISKS IN SEPA OUTPAYMENT SYSTEM BY USING DEFENSIVE TESTING

Bachelor's thesis

Supervisor: Martin Rebane
MSc

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Tatjana Kruglova 142442IAPB

RISKIDE MAANDAMINE SEPA VÄLJAMAKSESÜSTEEMIS ENNETAVA TESTIMISE ABIL

Bakalaureusetöö

Juhendaja: Martin Rebane
MSc

Tallinn 2017

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Tatjana Kruglova

21.05.2017

Abstract

The aim of the current bachelor's thesis is to identify, analyse, develop a test plan, and execute the test plan of the software changes needed for the Single Euro Payments Area (SEPA) regulation in Sweden. The adjustments made to Claim Outpayment Manager (OPM) must allow the customer to create payments and receive bank transactions in ISO20022 certified data format.

The thesis describes requirements to software according to SEPA regulation and possible risk factors that occur while implementing these requirements. Risk analysis was conducted to help identify potential threats and design corresponding test plan. Defensive testing was used to verify the readiness of the software and mitigate any risks involved in payments export and import of the bank return messages.

During the thesis work, requirements of SEPA regulation were listed down. Specifications were processed and potential risks were covered with negative test scenarios. Manual and automated test cases developed during current thesis work can be used for future testing of the OPM module.

This thesis is written in English and is 54 pages long, including 5 chapters, 13 figures and 14 tables.

Annotatsioon

RISKIDE MAANDAMINE SEPA VÄLJAMAKSESÜSTEEMIS ENNETAVA TESTIMISE ABIL

Lõputöö eesmärkideks oli identifitseerida, analüüsida, luua testiplaan ning testida SEPA määrusega kaasnevaid muudatusi Rootsis. Arendatud uuendused väljamaksesüsteemis peavad võimaldama kliendil luua ning vastu võtta makseid ISO20022 formaadis.

Lõputöö kirjeldab SEPA määrusega kaasnevaid nõudeid ning arendusega kaasnevaid võimalikke riskifaktoreid. Riskianalüüs on läbi viidud, et aidata välja selgitada võimalikke ohte ning selle järgi disainida testimise kava. Ennetava testimise meetod on kasutusel, et kinnitada tarkvara valmisolekut ning leevendada võimalike riske maksete ekspordil ning sissetulevate maksete haldamisel.

Töö käigus olid SEPA nõuded üles kirjutatud. Spetsifikatsioonid olid läbi töödeldud ning võimalikud riskid olid kaetud negatiivsete testilugudega. Töö käigus arendatuid manuaalseid ja automatiseerituid testijuhte saab tulevikus kasutada väljamaksesüsteemi mooduli testimisel.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 54 leheküljel, 5 peatükki, 13 joonist, 14 tabelit.

List of abbreviations and terms

API	Application programming interfaces
BBAN	Basic Bank Account Number
BIC	Bank Identification Code
Defensive testing	Form of negative testing, used to defend the system against improper usage.
EPC	European Payments Council
EU	European Union
IBAN	International Bank Account Number
ISO	International Organization for Standardization
ISO20022	ISO 20022 Financial Services Standard of financial industry message scheme [1]
IT	Information Technology
JavaScript	Programming language
Negative testing	Testing method to verify that systems performs correctly with incorrect inputs and activities.
OPM	Claim Outpayment Manager Module of Profit Life&Pension insurance product [2]
pain.001.001.03	Client-bank message format
pain.002.001.03	Bank-client message format
PGP	Pretty Good Privacy Encryption program [3]
PLP	Profit Life&Pension Software solution for insurance companies [4]
Positive testing	Testing method to verify that system performs correctly with correct inputs and activities.
Regular expression	A pattern that regular expression engine attempts to match in input text [5]
RIDE	Robot Framework Test Data Editor
Robot Framework	Framework that uses keyword-driven testing approach [6]

SEB Test Bench	Service to verify correct implementation of ISO 20022 standard [7]
SEK	Swedish krona
Selenium	Library written in JavaScript programming language
SEPA	Single Euro Payments Area European payment integration initiative [8]
SISU	Markup based document structuring [9]
SWIFT	Society for Worldwide Interbank Financial Telecommunication
UI	User Interface
XML	Extensible Markup Language Markup language for documents containing structured information [10]

Table of contents

List of figures

List of tables

1 Introduction

The Single Euro Payments Area (SEPA) was introduced to replace previous national payment service and develop a union in which companies and all the other economic parties are able to make and receive payment in euro under common conditions and in accordance to the same rights and obligations, regardless of their location [8]. The 31 October 2016 was a SEPA Regulation deadline in non-euro countries [11]. All payment service providers that offer payment services in euro must do so using same ISO20022 format [11].

Sweden is one of the few countries that allows to use the same regulation for their currency payments as well [11]. SEPA regulation and this decision have a great impact on companies. Crucial changes have to be considered by the businesses and software developers in order to deploy ISO20022 format changed into their business software.

The aim of the current bachelor's thesis is to identify, analyse, develop a test plan, and execute the test plan of the software changes needed for the SEPA regulation in Sweden. The adjustments are made to Claim Outpayment Manager (OPM), the system used by insurance companies to pay out claims and premium refunds [2]. The adjustments needed must allow the customer to create payments and receive bank transactions in ISO20022 certified data format.

This thesis focuses mainly on OPM SEPA formats:

- SEPA payments export
- Bank payment return for SEPA payments

Following key activities were pursued in order to fulfil bachelor's thesis objectives:

- Coordinate the adjustments to be made with the introduction of ISO20022 format
- Assemble analysis of the risks that are caused by the software update

- Develop a defensive test plan and number of test cases to help avoid aforementioned risks

- Implement the test plan and verify the software

Current thesis uses qualitative risk analysis and defensive testing to ensure the quality of created software. Qualitative risk analysis is created to review possible threats of developing the software update and determine risk reduction methods. The primary risk reduction method used in current work is the extensive use of defensive testing. Defensive testing is a form of negative testing, where the quality is ensured by validating the system's reaction against improper usage.

The result of the bachelor's thesis work is a software update that is used by the Swedish insurance company to send and receive payments in specified ISO20022 format that has undergone thorough risk analysis and profound defensive testing. Defensive test plan has been executed by manual and automated test cases created to validate the transition to a new format. Same test cases can be later used for regression testing.

2 SEPA regulation

The vision of the "Single Euro Payments Area" (SEPA) project was that citizens and businesses within one union should have an integrated market for electronic payments with no distinction between national and cross-border payments [8].

Before SEPA was introduced, customers struggled making euro payments within euro area, because these payments often required different formats and data elements that were costly and time-consuming [11].

To that end, the Single Euro Payments Area (SEPA) was introduced to replace current national payment service and develop a union in which companies and all the other economic parties are able to make and receive payments in euro under common conditions and in accordance to the same rights and obligations, regardless of their location [8].

This chapter has an overview of various SEPA payments methods, changes SEPA has brought to Sweden and therefore to the businesses.

2.1 SEPA payment methods

The SEPA project focuses on three payment methods: credit transfers, direct debits and card payments [11]. The European Payments Council (EPC) has established payment schemes for credit transfers and direct debits [11].

Credit transfer is a payment method where payer forwards payment instructions to their payment service provider and the provider moves the resources to the specified payee [11].

Direct debit is a payment transfer from payer's account initiated by the payee [11]. Direct debits are generally used for recurring payments and require pre-authorization by the payer [11]. This particular payment method had not been possible to use across countries before the SEPA regulation [11].

Both of the previous payment methods have a deadline of February 1st 2014 for the replacement in the euro area of national payments in euro created by Regulation No 260/2012 [11]. In countries that have other currencies, the deadline is 31 October 2016 [11]. As of these dates, existing retail payment schemes for payments in euro will have to be removed and replaced by SEPA alternatives [11].

Furthermore, the payers and payees cannot be restricted from a right to choose their payment account location [11]. Meaning that payers have a right to choose from which account in Europe they prefer to make their payments in euro and neither can payees be forced to accept payments in euro in an account held in a specific country.

2.2 SEPA changes in Sweden

Regulation No 924/2009[1] erases the charging differences for foreign and domestic payments in euro in all of the European Union (EU) Member States [11]. Regulation clearly states that all electronic payments, such as credit transfers or direct debits, cash withdrawals, and credit or debit card transfers have to use same charges regardless of their location [11]. Non-euro EU countries are authorized to apply this regulation also to their own currency. Currently Sweden and Romania have chosen to use this regulation [11].

Regulation No 260/2012[2] makes the use of certain standards and requirements mandatory to all EU Member States, requirements include the use of International Bank Account Numbers (IBANs) and financial services messaging standard ISO 20022 XML for all credit transfers and direct debits in euro [11]. Current regulation ensures the usage of previous Regulation No 924/2009 that introduced the principle of identical charges for all payment transfers made in euro [11].

Since Sweden uses kronas (SEKs) it was decided that they use Regulation No 924/2009 according to their currency as well. Which means that all the payments with all the

[1] "Regulation (EC) No 924/2009 of the European Parliament and of the Council of 16 September 2009 on cross-border payments in the Community and repealing Regulation (EC) No 2560/2001." [12]

[2] "Regulation (EU) No 260/2012 of the European Parliament and of the Council of 14 March 2012 establishing technical and business requirements for credit transfers and direct debits in euro and amending Regulation (EC) No 924/2009." [12]

different bank accounts: Swedish National Bank account, Bankgiro[1] number, Plusgiro[2] number, Foreign bank account, and International Bank Account Number (IBAN) use ISO 20022 format if needed.

This thesis takes a deeper look into Swedish interpretation of SEPA payments. Service providers support different payment types in Sweden:

- Giro payments (Bankgirot and Plusgirot) – Swedish administration of payment transfers
- Credit transfer with creditor bank account
- Same day transfer
- Salary payments
- Money order
- SEPA Credit Transfers
- SEPA Credit Transfers with same day execution

"One key area of impact for Swedish customers is changes that will take place within Bankgirot. During 2014 Bankgirot has, as a consequence of the SEPA End Date Regulation, decided to phase out domestic euro payment products (Leverantörsbetalningar euroo) within EU/EEA by latest 31 October 2016." [12]

2.2.1 International Bank Account Number (IBAN), International Organization for Standardization (ISO), and Bank Identification Code (BIC)

There are a lot of changes in business software needed with SEPA regulations. In order to understand the changes made, related terms need to be explained. First of all, previous bank account numbers need to be replaced with International Bank Account Numbers

[1] Bankgiro accounts are used by the credit agency Bankgirot of mass payments in Sweden [13].

[2] Plusgiro accounts are used by the PlusGirot money transaction system owned by the financial institution of Nordea [14].

(IBANs) and data formats are changed with global ISO 20022 message standards introduced by International Organization for Standardization (ISO) [13].

IBAN is an International Bank Account Number, an extended version of BBAN - Basic Bank Account Number, that identifies a unique bank account, at a specific financial institution, in a particular country [14]. IBAN follows an international standard for bank accounts called ISO 13616 [15]. SWIFT - Society for Worldwide Interbank Financial Telecommunication, a global provider of secure financial messaging services, has been nominated as the Registration Authority for ISO 13616 [15].

The structure of IBAN is defined in ISO 13616-1 and begins with two letters representing country code of the account-holding bank, followed by two numbers that are a control key, and up to thirty alphanumeric characters to for a BBAN which has a fixed length per country and includes a bank identifier within it [16]. For example, for Sweden the defined ISO 3166 country code is SE and IBAN total length is 24 characters, which means the IBAN in Sweden would look like "SE45 5000 0000 0583 9825 7466" [15].

The Bank Identification Code (BIC) is a standard to identify any beneficiary's bank account easily. The BIC system is administered by the SWIFT. BIC code is basically the same as the bank's SWIFT address [17]. The BIC length may vary from 8 to 11 characters. First four letters are usually shortened version of a bank's name, followed by two-letter country code and two letters or digits representing bank's head office location. Last three digits are optional to specify bank's branch code [18]. For the SEB bank in Sweden, it looks like "ESSESESS" [19].

Lastly, one of the most important changes for the software industry is the financial services messaging standard 'ISO 20022 XML'. This standard is defined by the ISO for the development of electronic financial messages [20]. Regulation states that physical representation of the payment transactions must be in Extensible Markup Language (XML) syntax and according to business rules and implementation guidelines of Union-wide schemes [21].

The changes affected different parties:

- Clients were the least affected, only thing that clients noticed is that the usual bank accounts format changed to a slightly longer format - IBAN.

- The private and public sector institutions had to accommodate their systems to use new SEPA payment formats.
- The banking sector had to implement the SEPA changes and bring them to the customers.
- The ministry developed and implement the legislation for these changes.

As all of the parties had to adapt to the new changes, they all had to cooperate. A company could not start their development until the legislation and banking decided what and how the changes need to be made. The banking industry was in the lead with this project. Banks did the interpretation of new legislation created by the ministry and the European Union laws, businesses in turn, had to adapt to the solutions banks made and wait for the solution to be testable.

Current paper's main focus is implementation of ISO 20022 standards in SEPA payment export (client-bank) and bank payment return for SEPA payments (bank-client) for Swedish life insurance company.

Looking into SEPA payment exports, there is no difference between banks. If solution works for example in Danske Bank then the same ISO 20022 – pain.001.001.03 format works in SEB bank as well. Whereas, each bank has its own payment return format for SEPA payments called pain.002.001.03.

2.3 SEPA changes in Claim Outpayment Manager

Claim Outpayment Manager (OPM) is one of the functionalities presented in Profit Life&Pension (PLP).

2.3.1 Profit Life&Pension

Profit Life&Pension (PLP) is a business product for life and pension insurers that includes aforementioned Claim Outpayment Manager, see Figure 1. The product has automated all key processes needed in insurance business, from product development to claim management [4].

Profit Life&Pension solution is built using service-oriented architecture and modular approach, therefore the web application programming interfaces (APIs) are easily

integrated with other applications and technologies [4]. This architecture allows each of the Profit Life&Pension functionalities to be used individually or to be implemented as a whole solution [4].

Profit Life&Pension supports insurers' key processes.

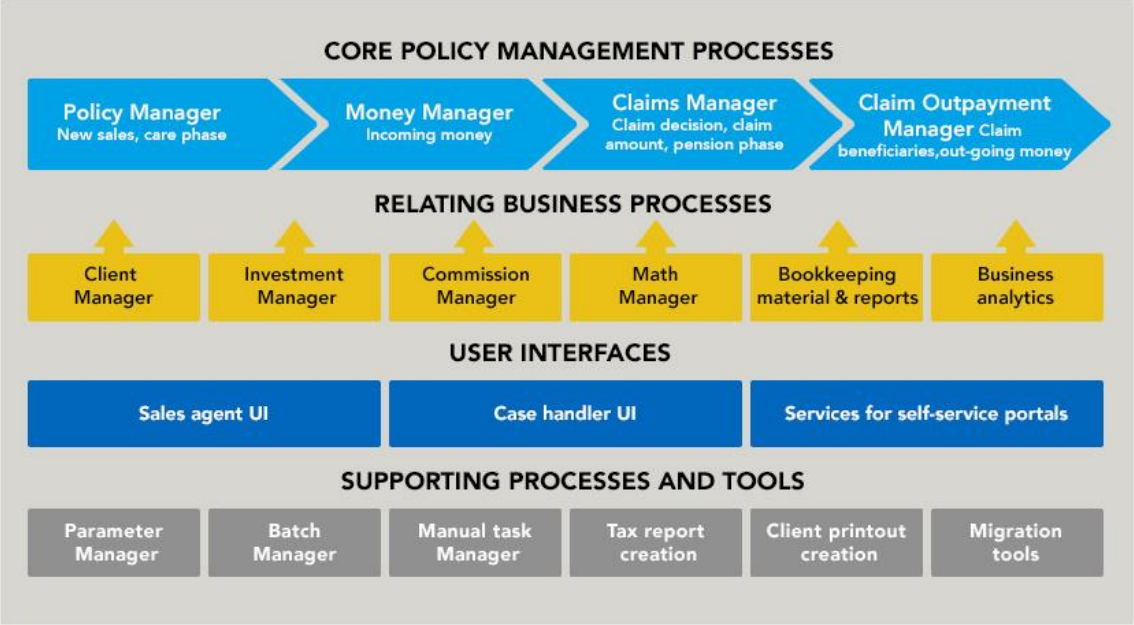

Figure 1. Modular structure of Profit Life&Pension [2]

2.3.2 Claim Outpayment Manager

As mentioned above, the Profit Life&Pension product is modular and therefore every module can be used separately in various projects. One of the key modules of the solution is Claim Outpayment Manager.

"PLP Claim Outpayment Manager is designed to pay out claims and premium refunds." [2] The process is independent from other managers, it only receives the monetary claim or premium return amount from the source systems [2]. In cases with recurring claims the system manages beneficiaries, beneficiaries' shares, payment receivers and claim reserves during the whole period [2].

Some of the key functionalities of the system is that to handle outgoing money export files to banks and banks payment return information [2]. In addition to that, the system also creates bookkeeping entries, printouts and their data, also internal reports and tax authority reports [2]. The whole process of claim and premium outpayment is

automatized otherwise the system writes a workflow messages to the users when the manual handling is required [2].

2.3.3 SEPA changes in Outpayment Manager

Before SEPA introduction outpayment manager followed scheme from Figure 2.

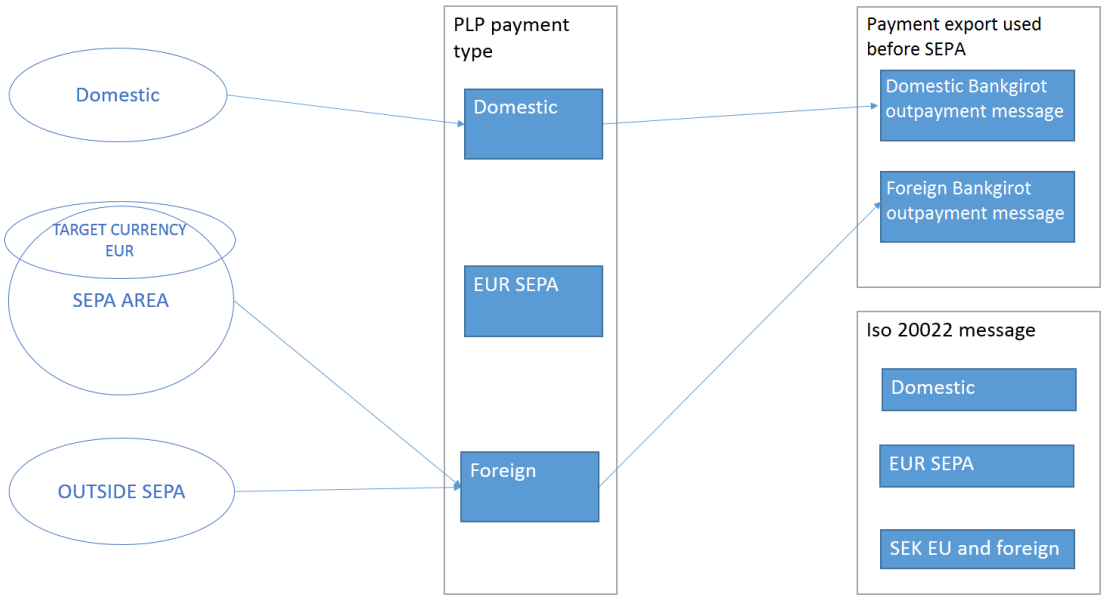

Figure 2. Outpayment export design before SEPA introduction (Author's illustration)

There were two requests from one of the Profit Software Ltd insurance company clients to change the system according to the new SEPA regulation.

Firstly, client specified that they need foreign payments, inside and outside SEPA area with payments types IBAN and Foreign non-IBAN, export to bank using SEPA file format. Coupled with handling the payment return information, payments confirmation or rejection, coming from bank using SEPA file format.

Foreign payments have to be delivered to bank using SEPA file format. Requested addition is the new batch which designs an XML file according to SEPA pain.001.001.03 schema. Same batch and file will be later on extended to handle domestic payments as well as the foreign ones. New elements and values are added with additional mapping.

Payments export functionality will remain resembling current payment export using Bankgirot domestic and foreign files. The payments will be collected to the separate table and later based on the table written into file. All the files will contain header, payer and

payee information. File will not contain Pretty Good Privacy (PGP) encryption signature because connection used is secure.

Additionally, the abroad and domestic payments will be held in separate files. File has specific name format and has maximum limit of 50 000 of payments per file, which is defined in process parameters. Handling more payments than defined limit will create multiple payment files.

Payment return information file from bank using SEPA file format needs and addition of another new batch which reads XML file in accordance with SEPA schema pain.002.001.03. Same batch will be later extended to administer domestic payments.

There are two return messages from the bank. First message will arrive with initial validations by the bank, checking the syntax and PGP signature. Second message will return after handling the payments. Second message corresponds to current message from Bankgirot to domestic payments. Return file must also have specific name format.

There are four types of return messages:

- Whole message rejection, all the payments are rejected.
- Message stopped, PGP signature is not valid.
- Confirmation of one payment.
- Rejection of one payment.

OPM handles confirmations and rejections on payment level, if the whole message is rejected or confirmed the OPM investigates payments one at a time and rejects or confirms them accordingly.

Conducting the feature request for the client challenges with the Swedish banking system occurred. The main issue was that Swedish banks do not offer any currency conversions; therefore, the payments should already be sent in euro and therefore be in euro in the source system. This means that if the client wants to go over to the SEPA regulation the current PLP should offer currency conversion and add all the interfaces for daily conversion rate values.

As a result of client request and finding that SEPA payments cannot be sent with SEK currency, the client has decided to use ISO20022 payment export format with European and foreign payments. Client requested adjustable control in order to select to which payment export each payment type is going to be sent. The payments are domestic and foreign. Additionally, the separation for SEPA payments is kept but changed to a subtype of foreign payments. SEPA is used only if the source and the target currency are EUR.

Parameter is added to select where are domestic and foreign payments being sent. Both batches, currently used and the new batch created for SEPA payments, are kept separate and the separate selection of handled payments will be based on the parameter added and rules to define domestic and foreign payments.

Foreign payments are payments with IBAN or foreign non IBAN bank account type or check payments and payment orders having target country other than the system country.

Payment is identified as SEPA payment if the payments bank account type is IBAN, payment source and target currency is EUR, and target country is defined in parameter as SEPA country.

Currently client does not have payments that fit these criteria, therefore SEPA payments are not actually used but there is a separation for those for later use. Currently payments will be classified only as European or foreign payments.

Domestic payments are defined as payments that are not foreign or SEPA.

There will not be separation for SEPA payments in current payment export using Bankgirot message formats.

The final new proposal followed scheme from Figure 3.

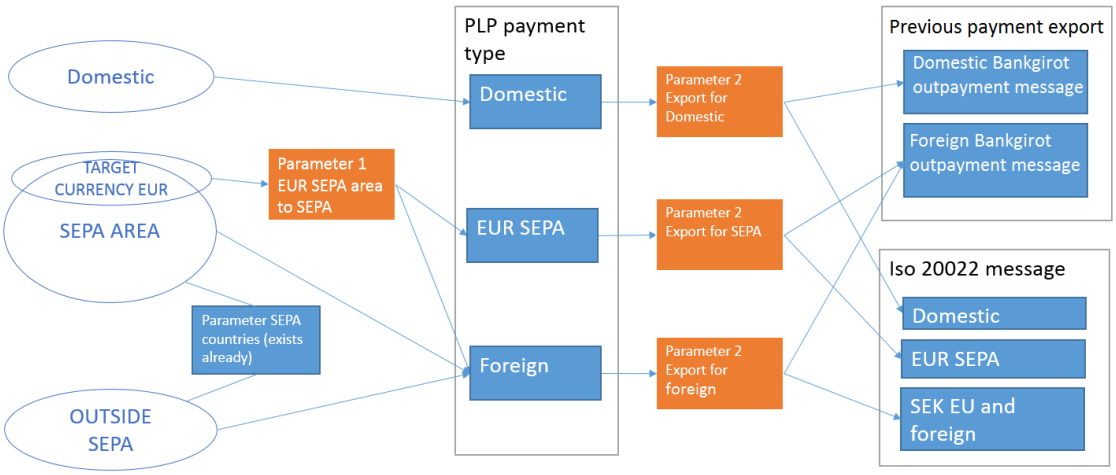

Figure 3. Outpayment export design after SEPA introduction (Author's illustration)

3 Risk analysis and quality assurance

Risks occur in every information technology (IT) project and risk analysis is needed to review of the risks associated with each a project or its' part. Risks are analysed on a quantitative and qualitative basis [22]. Risk analysis should be reviewed and updated on a recurring basis [22]. That way new potential risks are discovered as soon as they arise. Risk analysis is one of the components of risk management.

Risk management process involves identification and examination of potential threats [22]. For example, risks are associated with payments being sent to wrong bank accounts or using the wrong file format so the machine analysing the data on the bank side does not find it as a correct payment.

Risks are examined using quantitative and qualitative risk analysis [22]. Quantitative risk measures expected risk probability to forecast potential financial losses [22]. Qualitative risk analysis reviews threats, and determines and establishes risk mitigation methods and solutions [22].

Current paper focuses more on the qualitative risk assessment in order to identify and manage the risks to reduce the possibility of any financial loss. The way of defensive testing is used to eliminate as much risk factors as possible.

This chapter explains the definitions of quality assurance and software testing. Later on, testing methods are introduced and the usage of automated testing is explained.

3.1 Quality assurance and software testing

Software quality assurance is a set of activities to ensure quality in software engineering process [23]. The main idea behind it is to make sure that the product functions the way the client desires. Verifying the quality of the software relies strongly on testing of the software.

Software testing can be described as a process of validating software against business and technical requirements. Created software must satisfy client's needs and purpose. It is even more crucial to verify that the software is able to continue functioning in abnormal situations. Therefore, testing is often portrayed as a means of exposing faults in the system

[24]. “Finding the defects helps us understand the risks associated with putting the software into operational use, and fixing the defects improves the quality of the products.” [24] Nevertheless, detecting defects is only a part of the testing process as a whole.

Test design must be thorough. One of the testing fundamental principles is that we cannot test all the possibilities [25]. Projects simply never have enough time and resources to do so. As a result, test focus is decided based on the critical risks that have been assessed in the risk analysis. There are several methods to ensure the quality of the software is good, risks are minimized and the resources not exhausted.

3.1.1 Equivalence partitioning

Testing different possibilities without wasting billable time is one of the key features in testing. The main idea is to divide test input data into several classes and selecting one input value from each class is sufficient to verify the whole class [26]. This method helps to reduce an infinite number of test cases to a finite number, ensuring that the selected test cases are still effective and cover all possible scenarios [26].

Let’s assume we have a numeric input field that only accepts values from 0 to 10. Using equivalence technique, we can divide inputs in three classes:

- Valid input from 0 to 10
- Invalid input below 0
- Invalid input above 10

After the classes have been chosen we create one test case per class. In this example, we would have one test case for a valid input, where we select any input from 0 to 10. This shows that we do not need ten test cases to check every valid input, one should be enough to cover the functionality. The remaining two test cases we need to verify that the invalid inputs are not accepted by the system. We create one test case where the input value is below the lower limit i.e. any number below 0 and second one where the input value is above the upper limit i.e. any number above 10.

The reason why we can choose one value from 0 to 10 and still test all the possibilities is that programmers use similar technique in their code. In most programming languages, this kind of check is performed by an if-clause.

3.1.2 Boundary value analysis

Although, usually the errors are present not in the three classes described above but in the very extreme ends of the input values [26]. So, in this case in addition to checking the values from the middle of our class, we check the boundaries of the input range.

Equivalence partitioning and boundary value analysis are usually used together in testing [26]. Edges of the equivalence classes create test cases for boundary value analysis [26].

Each boundary value is either valid or invalid and test cases must be created for both valid and invalid values [26]. Which adds our input field another six test cases:

- Test cases 4 and 5 for both valid boundaries, 0 and 10
- Test cases 6 and 7 for both values adjacent to lower boundary 0, -1 and 1
- Test cases 8 and 9 for both values adjacent to upper boundary 10, 9 and 11

So, in order to test one input field, there should be at least 9 test cases to ensure that its behaviour is correct in any circumstance.

3.1.3 Positive and negative testing

This same technique explained above can be looked at a different angle and generalized into two categories: positive testing paths and negative testing paths [27].

Positive testing is a sort of testing that every tester performs as a first thing on an application [27]. Positive testing means that the tests are run exactly like the end user would use the application, therefore tester only works with the correct data input and ensuring that the product has met the requirements [27].

Usually there might be many alternate paths to verify the same requirements and all of these variations are also part of the positive testing, as long as all the inputs are valid and correct [27].

As opposed to the positive testing routine, testers also have to validate that the product can cooperate with invalid data and activities. In this case, the negative testing is performed to validate the product against any invalid inputs.

Negative testing is important to ensure that the end user is shown correct errors when it is appropriate and the system does not stop working or give wrong results when the faulty data is used [27].

3.1.4 Defensive testing

Current thesis uses the negative approach to define defensive testing. Similarly to defensive programming defensive testing is used to ensure software can continue functioning under unforeseen circumstances [28].

3.1.5 Must-do negative test cases

According to Steve Miller, there are at least 10 negative test cases that should be considered when designing the test effort [29]. Negative cases used in current thesis are explained further.

1. Single quote insertion – entering text input with single quote might reveal database issues [29].

2. Required data entry – specifications always contain the fields that are require data entry before continuing [29]. It is important to try and leave empty the required fields in order to ensure that the application forces to enter the data into the required field.

3. Field type test – specification also defines fields that require specific data entry (numeric fields, date fields, personal identification numbers, etc.) [29]. It is crucial that these fields that require special types are verified.

4. Field size test – functional specification has all the fields sizes defined and most of the database systems have predefined field size [29]. Each field is allowed specific number of characters (for example, phone number must be 10 characters or less), therefore the tests must verify that field only allows to input specified number of characters.

5. Numeric bounds test – numeric fields should be tested for their lower and upper bounds [29].

6. Date bounds test – date fields should also be verified against their lower and upper bounds [29].

7. Numeric limits test – numeric fields might not have specified the lower or upper bound, yet most of the databases have minimum and maximum bounds defined by the type chosen (from small integers to big integers) [29].

8. Date validity – all input date must be valid dates with valid format [29].

9. Web session testing – most pages cannot be seen to unknown user; therefore, browser is intrusted to validate correct users [29].

10. Performance changes – speed of the application should be deemed tolerable [29].

There are a few more test cases that author of current thesis thinks should be considered in addition to the previous suggestions:

11. Correct user tests – different users might have different rights within the application. Tests should be created to verify user rights and visibility.

12. File input tests – applications deal with the information coming in or leaving the system as some sort of a file, yet files are easily corrupted. If the application allows input files, there must be a few tests that verify the structure of the file and as in the previous cases ensure that the rules for the required, specific format and sized fields are also obliged.

13. Injections tests – application must also cope under the deliberate attack.

14. Concurrent usage of the database – most of the systems have many users simultaneously doing changes in the system. There is a possibility that several of the users are applying changes in the same area of the software, therefore updating the same database table or even row. These updates must be controlled in order to eliminate the possibility of corrupted data.

15. File split limit check – systems can be immense and thus the data coming in/leaving/being stored in the system is truly large. Having system exporting or importing large files can be unreliable. A method of splitting the file into smaller pieces is often used in this situation. Having files split must also be verified that no information goes missing in the process.

3.1.6 Regression testing

In addition to positive and negative testing of the feature, the crucial part of the quality assurance is to verify that the previously developed and tested features are still performing correctly after introducing the new functionality. The Claim Outpayment Manager is strongly reliant on test automation to verify the regression issues.

3.1.7 Test automation

Seeing that there are so many tests needed for even a small functionality, we could barely retest everything every time implementation changes are made. Automating previously created tests, helps with regression and gives developers the confidence to do the bigger changes.

Automatic test case is the same test case rewritten in a programming language. The script consists of the steps, input data, validations and verifications and the tests results.

Automatic tests help to increase testing result with limited time and resources. Tests that are time consuming and might normally take hours to go through manually, can be done in minutes using automated test scripts.

Building automated tests is challenging, especially for the web applications [30]. The user interface of the application constantly changes, there might be incompatibilities between browsers or browser versions, and most commonly there is a need to support various server and client platforms [30]. Therefore, there are tools made to ease building and executing automated test cases for applications [30].

3.1.8 Test automation tools used

In current thesis work and project the help of Selenium and Robot Framework was used to create test automation. Test automation is not a primary target of current work, but it is essential for OPM's further development.

Selenium is a library written in JavaScript programming language that allows different interactions within a web application. It can be easily used in unit tests as well as in user interface test because it is not tied to any particular framework [31].

Since Selenium acts as a library, Robot Framework is used as a binding framework. Robot Framework is built on Python and is application independent [6]. Robot Framework has

various test libraries that are ready to use, Selenium included, while each project can extend the framework with libraries using either Java or Python [6].

Robot Framework uses keyword-driven testing approach and can be easily used to write data-driven tests. If testing team does not have a programming background then using keyword-driven method is easy to learn and users can create new higher-level keywords from existing ones also using the same syntax as the test cases [6]. Most importantly, Robot Framework provides its users with clear reports and detailed logs.

During this thesis work, Selenium and Robot Framework test automation tools are used with the help of Robot Framework Test Data Editor (RIDE). Test automation is used to verify implementation changes quickly and test for any regression issues. Regression issues are tested with the help of OPM's previously developed automated test cases.

4 Identifying and testing process

Current chapter contains descriptions of identified risks found during the risk analysis and profound testing process and it's results are explained. Additionally, the comparison between previous and SEPA payment exports is brought out and regression testing is conducted.

4.1 Identifies risks

As a result of thesis work two core areas of migration risk were found. In addition to the business risks themselves, one more risk area occurred during the developement.

4.1.1 XML

The standard XML format was created by the ISO to ensure that the bulk payment files are correctly formatted and content is sufficient to allow the company, bank and the beneficiary to maintain processing and reconciliation levels.

Formatted XML files have their own components according to the payment types, their priority and payment methods.

Regarding ISO20022 format, it is crucial to verify all the file formats with various client and payment types are according to the service provider scheme. SEB Test Bench service to verify the correct implementation of ISO 20022 standard is used to evaluate file format created by the outpayment system [7].

4.1.2 IBAN and BIC

Making sure that on all payments are correctly identified the bank account using the BIC and IBAN. Any organization in the SEPA is legally required to include BIC and IBAN on invoices. Therefore, it is time to make sure that all of the suppliers, employees and payment beneficiaries have provided the required information.

The use of BIC and IBAN information is now mandatory for domestic payments in addition to cross-border payments. The use of IBAN/BIC resolves problems with payment efficiency and some technical issues [32]. In Sweden, some banks'

(Handelsbanken) IBANs do not contain domestic bank code. In order to reflect bank code, the addition of zeros is used [32].

For example [32]:

Bank Code: 6041

Account: 014086998

Relates to IBAN: SE04 6000 0000 0001 0408 6998

The challenge itself is to collect the information for all the companies, beneficiaries and receivers. It is also important to verify the IBAN and BIC compatibility. Incorrectly formatted IBAN and BIC pair will result in payment rejection.

In this work, OPM is dependent on the users and source systems. Bank accounts are managed either manually or imported from other systems. The only verification that can be made is the IBAN and BIC code format validity.

4.1.3 Parametrized implementation

Client is allowed to use the version they prefer and change the parameters without any limitations. The risk of faulty settings is present. To prevent adjustment issues the testing of the software must focus on parameter value changes.

One of the methods to manage aforementioned risks is to create tests based on risk factors. Testing helps to measure the quality of software in terms of the number of defects found and functionality covered. A well-designed test set will uncover possible defects and overall level of risk will be reduced.

4.2 Comparison of payment exports

As agreed with the client, there will be three types of payment exports using ISO 20022 format shown in Figure 4.

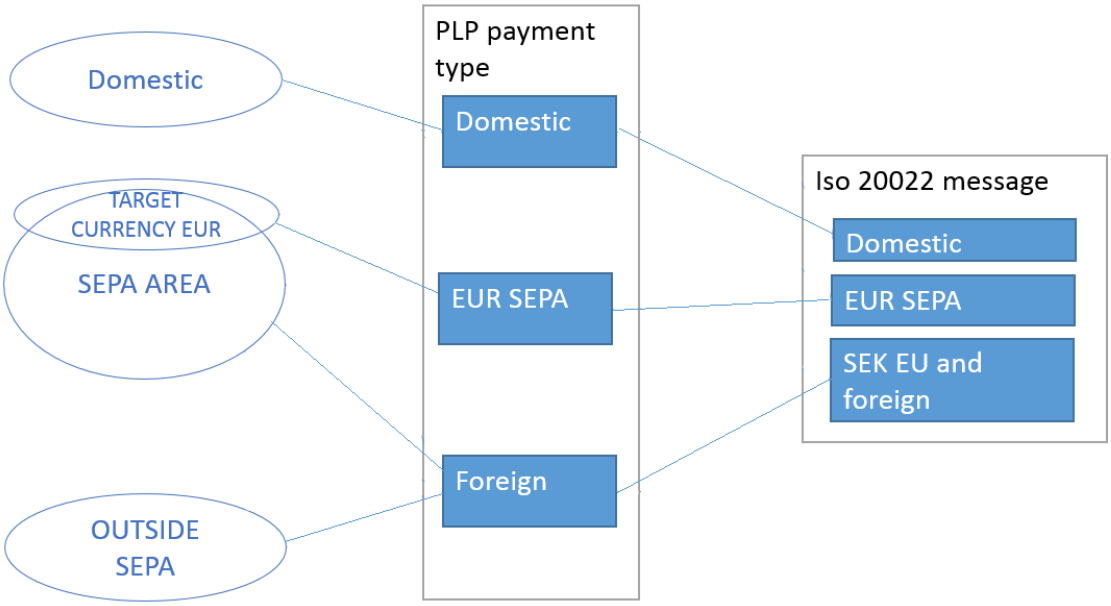

Figure 4. ISO 20022 messages (Author's illustration)

Collected payments needed to be exported via ISO 20022 message are verified against certain rules. Payments that are exported should be in accepted status and payment should not be of internal type – this functionality was already in place before SEPA introduction.

After collecting all the external accepted payments, they are divided into three parts: domestic, foreign, and SEPA. Foreign and domestic payments regulations were also in place before the SEPA change. Addition created by the OPM developers is that now the foreign payments with currency as EUR are defined as SEPA payments and use SEPA export file.

Foreign payments are ruled as payments that have bank account type of IBAN or Foreign non IBAN or all the payments where the payment methods are classified as check or payment order and having payment country different from Sweden.

SEPA payments must contain IBAN account from SEPA area and use target currency as euro.

Domestic payments are classified as all the payments that have any other bank account than IBAN or Foreign non IBAN and all the payments orders and check payments that have Sweden as their payment country.

Each payment should be handled once by the batch, after the payment has been collected to the export file, payment goes to paid status and is not collected by the batch any more. After that paid payment waits for verification from the bank.

4.2.1 SEPA payment export file format

OPM uses SEPA export file according to introduced pain.001.001.03 standard.

File names begin with a prefix which is one of the following "abroad", "sepa" or "domestic" followed by a timestamp which is payment exporter processor run time.

The payment initiation message is composed of three main building blocks:

- Group header – group header is mandatory and present once at the top of the file. Header contains common elements to the entire message
- Payment information – payment information is mandatory and repetitive block. This is the information about elements related to the debit side of the payment. Each block represents one payment and payer information
- Credit transfer transaction information – component of the payment information block. This information is also mandatory and repetitive. This is the information about the credit side of the transaction with receiver and amount information. Each block represents one payment and payee information.

Group header has several important fields to be verified during the testing process, Table 1 illustrates fields that have been covered with defensive testing during current thesis work.

Table 1. Group header format

Message Item	**Definition and mapping to OPM**
> Group header	
>> Message identification	Only allowed characters are according to the regular expression: `<xs:pattern value="([A-Z,a-z,0-9]|[\+|\?|/|\-|:|\(|\)|\.|,|'| ]){1,35}"/>` This value is unique for 60 days and is used to unambiguously identify the message by bank, in order to verify for the duplicate messages.

>> Number of transactions	Number of individual transactions contained in the message. The number of transactions must be equal to the number of Credit transfer transaction information payments (third level of the message).

Within the payment information structure Table 2 presents differences between domestic and foreign payment exports and common transaction fields.

Table 2. Payment information format

Message Item	**Definition and mapping to OPM**
> Payment information	
>> Payment information identification	Payment reference assigned by the OPM.
>> Payment method	Payment method field can have two values. If the payment method is chosen as check payment or payment order, the field is occupied with constant "CHK". Using other payment methods will add constant "TRF" to this field.
>>> Service level	
>>>> Code	Constant "SDVA" is present when system country is defined as Finland and payment method is not check and due date of the payment is smaller than the current date. Constant "NURG" is defined by Swedish domestic payments with payment methods as payment order or bank account is either Bankgiro, Plusgiro or Swedish bank account. If none of the conditions above are true, "SEPA" is set as the payment code
>>>> Proprietary	Current field is only present for Swedish domestic payments, it is determined by the constant "MPNS".
>>> Local instrument	Field is not used if the service level fields are already present.
>>>> Code	Additional code constant "IN" is added for check payments and "SDCL" constant is used for same day payments.
>> Debtor account	
>>>> IBAN	Used for foreign payments with IBAN bank account or Foreign non IBAN bank account.
>>>> Other	Structure is required if IBAN is not used. Used for domestic payments having Bankgiro, Plusgiro or Swedish bank accounts.

Credit transfer information structure is critical to risk management is defined in Table 3.

Table 3. Credit transfer transaction information format

Message Item	Definition and mapping to OPM
>> Credit transfer transaction information	
>>> Amount	Payment reference assigned by the OPM.
>>>> Instructed amount	Transaction currency and amount. Decimal sign is used as dot not comma.
>>>> Currency	ISO currency code is used by the amount's currency.
>>> Intermediator agent	Separate structure for foreign payments to Russia.
>>> Creditor agent	Created for all payments id payee's bank account information is found.
>>>> BIC	BIC code is present for foreign payments.
>>>>> Clearing system member identification	Used for domestic payments.
>>>>>>>> Code	Bankgiro and Plusgiro bank accounts have SESBA as clearing system identification and other banks have first five characters from the clearing code. Clearing codes should be created from old FIN-codes. For example: Old FIN-code -> New ISO 20022 AU123456->AUSBS/123456 BL12345678->DEBLZ/12345678 CC123456789->CACPA/123456789

This block also contains information about the receiver. Receiver name, address, bank account, reference numbers which is not presented in the Table 4.

4.3 Bank payment return for SEPA payments

In addition to payment export the system needs to recognize the bank payment return information for SEPA payments.

The batch created handles return information from bank about successful and rejected SEPA payments.

There are two messages that bank sends back to the clients. First message comes from the bank after the initial validation for syntax and PGP signature. Second message is received after the sent payments have been handled.

Firstly, system validates the input file. Validations include verifying that input file is in the designated location, verification on whether the input file has already been processed, therefore each file has unique name and timestamp in the header and if the file format is according to the input file description.

Later, the system handles rejections. Payments can be rejected on two levels. Group status rejection means that all the payments in the original payment export file have been rejected. Payment information status rejection shows that all the payments in payment block have been rejected. System then manages the rejected payments. Only paid payment status allows returning of the money, in other cases the new payment correction is created.

Lastly the accepted payments are attended to. On-going recourse payments are recourse and other paid payments are now confirmed by bank.

4.3.1 Bank payment return import file format for SEPA payments

Similarly, to the SEPA export files the bank payment return files also include group header, payment information and status and transaction information and status blocks. In addition to that, there is also a group information and status block added.

In group header section, there are only sender identification fields like banks' BIC code.

Group information and status message fields show whether the original export file has been rejected or accepted with rejection code and reason if needed.

Payment information and status section refer to each sent payment information block. Payment status can be acknowledged for the whole or only the parts of the section or rejected for the whole section. Additionally, status field can contain a request for execution date change.

Transaction information and status fields can also be either rejected or acknowledged. This section also contains information about original transaction from the export file,

including amount and currency, bank account number and creditor name and bank account.

4.4 Testing process

During current thesis work substantial amount of defensive test scenarios, described in Table 4, were created and executed to ensure the quality of the feature and reduce risks in production. Each scenario has several tests created and each functionality within the scenario is marked either successful (OK) or not successful (NOK).

Table 4. Defensive scenarios to be validated against the system

Defensive scenarios	Description
Parameter edge cases	Two parameters each have two possible values. Each parameter regulates several payment methods and bank accounts.
Incorrect IBAN and BIC	System accepts bank accounts as user input and from source systems.
Incorrect incoming file format	Incoming files can be corrupted and have several fields missing or incorrect.
Incorrect incoming file data	Incoming files can contain incorrect data.
Duplicate files	Incoming files can come in as duplicates.
Incorrect export file format	ISO20022 file format must be correct with mandatory fields and data for each file type.
File split limit	System can have substantial amount of payments to exported. Files are split according to the payments number.
Concurrent usage	Several users can use the same client or payment simultaneously.

4.4.1 Parameter edge cases

One of the most important additions that had to be tested was the parameter configuration. There are two parameters, shown in Table 5, one for domestic and another for foreign payments. Both parameters have two options, to be exported as normal payment (Bankgirot file formats) or in ISO20022 format. Parametrization is displayed in Figure 3.

Table 5. Parametrized payment export values

Payment type	Export type
Domestic (10)	Normal (10)
	ISO20022 (20)
Foreign (20)	Normal (10)

	ISO20022 (20)

To verify the parametrized export we need at least 4 tests to verify that all the payments are exported in correct format under the correct parameter.

Payments to be exported must be in accepted status and payment type is any other than the internal payment. The distinguishing of domestic and foreign payments is shown in Table 6.

Table 6. Domestic and foreign payment types

Domestic payments	**Foreign payments**
Swedish National Bank account	IBAN bank account
Plusgiro bank account	Foreign non IBAN bank account
Bankgiro bank account	Check payment with payment country any other then Sweden
	Payment order with payment country any other then Sweden (not possible in OPM)

Since the verification of bank accounts is crucial, all the possibilities of those must be included in the testing.

Testing done contains all six possibilities and is tested with all four parameter values. Created tests have combined over 24 possibilities from different client and bank account types to payment types. Testing has ruled that all classifications of payments are working correctly and each payment is exported to correct format.

4.4.2 Incorrect IBAN and BIC

There are several options on how the client can be added to the system.

Adding client from the user interface there must be verifications on IBAN and BIC formats. User interface (UI) should prevent the possibility of adding the incorrect bank account. The structure of IBAN is defined in ISO 13616-1 and begins with two letters representing country code of the account-holding bank, followed by two numbers that are a control key, and up to thirty alphanumeric characters to for a BBAN which has a fixed length per country and includes a bank identifier within it. The BIC length is may vary from 8 to 11 characters. First four letters are usually shortened version of a bank's name,

followed by two-letter country code and two letters or digits representing bank's head office location. Last three digits are optional to specify bank's branch code.

Client can also be added to the system via integrations. Integration is done using XML file with client information and run by the file processor. Handling this XML file, verifications on client bank account must be created. Moreover, if the client with incorrect bank account already exists in the system, the payment export should not allow payment to be processed without valid bank information. If payment method is set as check payment or payment order, then bank account information should not affect the payment export.

Testing process of IBAN includes mainly three types of tests by adding incorrect IBAN using non-existent country code, wrong length of BBAN code, and incorrect control key. Each of the test cases resulted in error of wrong bank account format and incorrect account number was not saved, example error is shown in Figure 5.

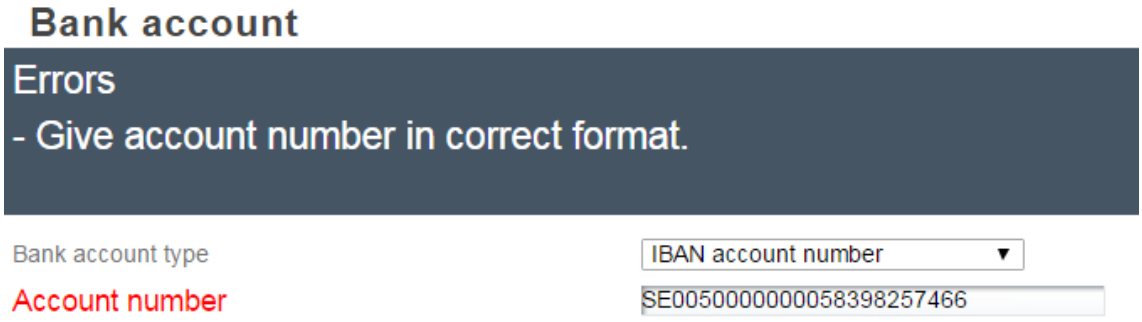

Figure 5. Incorrect IBAN account notification

Similarly, negative cases were created to verify the addition of BIC codes using invalid input length. Considering that BIC codes are not that strongly regulated the input itself cannot be tested besides the length. The BIC code with flawed length is not accepted by the system and error shown in Figure 6 is presented to the user.

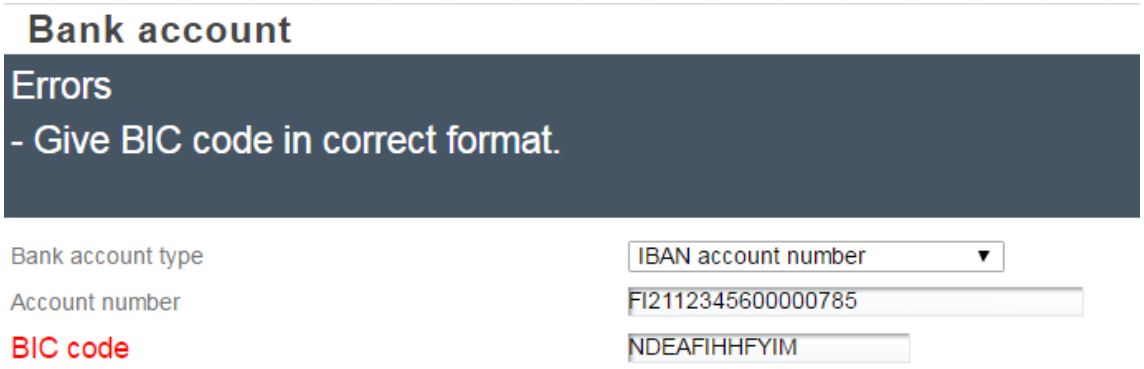

Figure 6. Incorrect BIC code format notification

In addition to UI tests, same verifications are made with client import files. New clients can be added to the OPM system via XML formatted file and the file processor must validate both IBAN and BIC.

As a result of file inputs with incorrect IBAN or BIC, the file processor handles the input files and errors are presented to the user. Clients added during this process are not saved and do not appear in the system.

Final tests results included 12 test cases and their status is described in Table 7 below.

Table 7. IBAN and BIC format test scenarios

Tests	Number of test cases	Status
IBAN input from UI	4	OK
BIC input from UI	2	OK
IBAN input from XML	4	OK
BIC input from XML	2	OK

4.4.3 Incorrect incoming file format

All incoming files in OPM are handled by processors. Each incoming file must be added to designated location and have a unique file name. Bank payment return file must contain four sections and each section must have correct fields.

File must begin with group header which contains message identification, creation time and date and information about initiating party. Example of the file group header is in Figure 7.

Figure 7. Group header information

Second block is original group information and status. This section contains original message identifications, group status and status reason. Figure 8 has group information and status example.

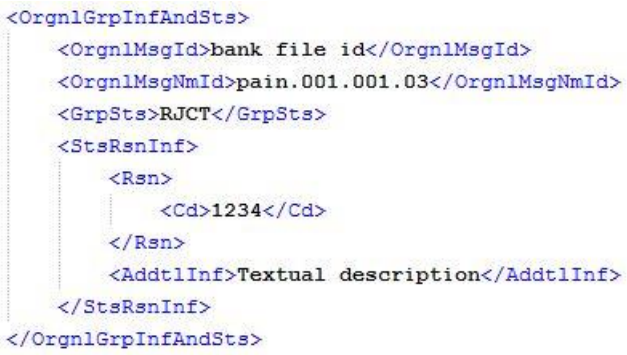

Figure 8. Group information and status

Third block has information about original payment information and status. This also contains identifications about original export file, control sum, payment information status and its' reason.

Last block shows the transaction information and status. The information about status identifications is also from the sent file and transaction reason and status are added by the bank.

Payment information and transaction information with their statuses are displayed in Figure 9.

```
<OrgnlPmtInfAndSts>
    <OrgnlPmtInfId>PAYMENTTKC3</OrgnlPmtInfId>
    <OrgnlCtrlSum>144.00</OrgnlCtrlSum>
    <PmtInfSts>ACSP</PmtInfSts>
    <!-- C. Transaction information and status -->
    <TxInfAndSts>
        <StsId>transaction id</StsId>
        <OrgnlInstrId>PAYMENTTKC3</OrgnlInstrId>
        <OrgnlEndToEndId>PAYMENTTKC3</OrgnlEndToEndId>
        <TxSts>RJCT</TxSts>
        <StsRsnInf>
            <Rsn>
                <Cd>1234</Cd>
            </Rsn>
            <AddtlInf>Textual description</AddtlInf>
        </StsRsnInf>
        <OrgnlTxRef>
            <Amt>
                <InstdAmt Ccy="EUR">144.00</InstdAmt>
            </Amt>
            <ReqdExctnDt>2016-05-10+03:00</ReqdExctnDt>
            <Cdtr>
                <Nm>IBAN-FI, IRENE</Nm>
            </Cdtr>
            <CdtrAcct>
                <Id>
                    <IBAN>FI1110465664298091</IBAN>
                </Id>
            </CdtrAcct>
        </OrgnlTxRef>
    </TxInfAndSts>
</OrgnlPmtInfAndSts>
```

Figure 9. Payment information and transaction information

Status reasons must be added to these fractions when message status is rejected. Having rejected status without reason specified results in an error created by the file processor. Figure 10 has an example of group payment information status as rejected and no reason added for the rejection.

```
<!-- B1. Group info & status: Accept or reject all payments -->
<OrgnlGrpInfAndSts>
    <OrgnlMsgId>bankfileid</OrgnlMsgId> <!-- bankfileid -->
    <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
    <GrpSts>RJCT</GrpSts>
    <!-- StsRsnInf: Must be used when status is RJCT -->
</OrgnlGrpInfAndSts>
```

Figure 10. Group message rejection without reason

Accepted payments do not have reasons identified.

Several tests were also created to verify the validity of the file with some of the file sections missing.

Test results are as in Table 8. Unsuccessful test case is created when accepted payment with return reason available is validated in the system.

Table 8. Bank payment return file format test scenarios

Tests	Number of test cases	Status
Group header missing	1	OK
Original group information missing	1	OK
Original payment information missing	1	OK
Original credit transfer missing	2	OK
Rejection without reason	3	OK
Acknowledgement with reason	3	NOK

Further investigation of having return reason available with accepted payments suggested that this is not a critical issue and will not be validated in the system. It is assumed that bank provides correctly formatted return files and return reason filed itself does not add any value to the accepted payment.

4.4.4 Incorrect incoming file data

Bank payment return file must also contain correct information in the fields described above. All original field values must be the same as in sent payment export file.

Testing revealed that incoming file message identification or transaction identification are not validated. Payments are updated according to their payment identifications. Message identification is used only in the case of group rejection. File with invalid message id and group rejection status is ignored. File is not processed and issue is not

recorded. Having bank payment return with payment id that does not exist in the system results in error in processor log, error states that original payment is not found.

Table 9. Bank payment return file data test scenarios

Tests	Number of test cases	Status
Original message identification is invalid	1	NOK
Original payment information is invalid	1	OK
Original transaction identification is invalid	2	NOK

As in previous testing, issues in incoming bank return files, seen in Table 9, will not be addressed by the system. Received files are considered to be correctly formed with authentic information.

4.4.5 Duplicate files

Duplicate incoming files are distinguished by unique file name. Also, if the payment has already been processed once according to the return file, then the same payment cannot be processed again. If the payment had been accepted, then the payment status is already confirmed by bank and status cannot be changed. In rejection cases, the correction payments have been made and original payment has been invalidated.

Re-running same file with same name shows information about input file being already processed. Files with same data but new file name does not produce any errors and files are processed by the system. Payments are not updated according to the second duplicate file, because all the files have been updated previously, therefore statuses have already changed and corrections made. As Table 10 shows, re-processing same file does not produce any issues in the system.

Table 10. Duplicate bank payment return file test scenarios

Tests	Number of test cases	Status
Same file name	1	OK
Different file name	1	OK

4.4.6 Incorrect export file format

Running ISO20022 payment export according to the parameters should create export file according to the payment type, bank account and payment method. All mandatory fields should be present and filled with valid information.

Domestic and abroad payment export files are created with correct message structure. Message contains group header, payment information and credit transfer transaction information sections.

Message identification must follow regular expression rules defined in the ISO 20022 standard:

```
<xs:pattern value="([A-Z,a-z,0-9]|[\+|\?|/|\-|:|\(|\)|\.|,|'| ]){1,35}"/>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>Sepa-70-20160517_124348</MsgId>
            <CreDtTm>2016-05-17T12:43:48.993+03:00</CreDtTm>
```

Figure 11. Inaccurate message identification format

Underscores (Figure 11) are not allowed by the regular expression pattern.

Domestic payments using all three Swedish bank accounts pass all validations. Files are created correctly.

Abroad payment using IBAN bank account and Swedish kronas are exported properly. Client information and bank account are verified.

Specification and implementation errors were found in foreign payment bank account presentation. ISO20022 scheme for foreign non IBAN creditors must contain constant BBAN as its proprietary and the bank account must be named as “Foreign non IBAN”. In addition to that, if foreign bank account does not have a BIC code, the clearing system member identification is not created (Figure 12). Identification field must be created for abroad payments not using IBAN bank account.

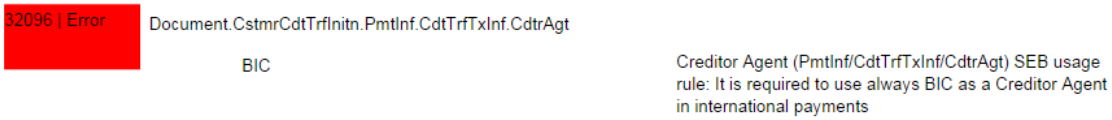

Figure 12. SEB Test Bench validation error for BIC code

Creating SEPA payment export with IBAN bank account and currency set as euro in order to test not only foreign and domestic exports but also the possibility to send actual SEPA exports. Executing payment export process results in technical error and no export file is created. After first issues is corrected, the SEPA file is missing creditor agent fields.

Check payments to abroad country have passed validations. Payment orders to abroad country are missing a local instrument code.

Abroad Russian payments and same day payments are processed as HIGH priority and are validated successfully.

Table 11 results are tested using SEB Test Bench validator.

Table 11. Payment export file format test scenarios

Tests	Number of test cases	Status
Message structure	3	OK
Message identification	1	NOK
Domestic Bankgiro payment	2	OK
Domestic Plusgiro payment	2	OK
Domestic Swedish BBAN payment	2	OK
Foreign payment with IBAN	2	OK
Foreign payment without IBAN	2	NOK
Foreign payment without BIC	1	NOK
SEPA export with currency as euro	1	NOK
Foreign check payment	2	OK
Foreign payment order	2	NOK
Abroad Russian payments	1	OK
Urgent payment	1	OK

Re-testing problematic areas of the file leaves no more issues open as presented in Table 12 below.

Table 12. Retested payment export file format test scenarios

Tests	Number of test cases	Status
Message structure	3	OK
Message identification	1	OK
Domestic Bankgiro payment	2	OK
Domestic Plusgiro payment	2	OK
Domestic Swedish BBAN payment	2	OK
Foreign payment with IBAN	2	OK
Foreign payment without IBAN	2	OK
Foreign payment without BIC	1	OK
SEPA export with currency as euro	1	OK
Foreign check payment	2	OK
Foreign payment order	2	OK
Abroad Russian payments	1	OK
Urgent payment	1	OK

4.4.7 File split limit

PLP system handles file split limits with parameter. Parameter value shows how many maximum entries export file must have. Creating more payments than parameter value should create more export files. Each file must contain maximum number of payments and the last file is with extra payments. Each file format must be correct.

Table 13 shows that single test is created to verify file splitting. Testing starts with changing file split limit parameter to three. By default, this amount is too big to verify all the payments are in correctly in files. Therefore, test is made with smaller number and with ten different payments. Creating ten payments eligible for payment export and running payment export process will result in four files. Three of the files must contain each three payments and last file should contain the tenth exported payment. Each file is made according to the export file format.

Table 13. Payment export file split limit test scenario

Tests	Number of tests	Status
Three entries per file	1	OK

4.4.8 Concurrent usage

If several users or processes update the same payment or client and database update fails due to concurrent usage, then payment or client handling should be skipped and all changed made should be reverted to their previous state. If handling has been skipped by the processor then the user must be informed and make the correct changes manually.

Testing (Table 14) is done by editing payment manually, this locks the database rows connected to the payment. At the same time payment export or bank return processor is run with the same payment information. File processor skips handling of the payment and restored database changes made during the batch run. In case of payment export, the payment will be processed by the next batch run. In case of the bank payment return, the original payment has to be updated manually.

Table 14. Payment handling concurrent usage test scenarios

Tests	Number of test cases	Status
Incoming file processor with manual usage	1	OK
Payment export file processor with manual usage	1	OK

4.5 Regression testing

Prior to sending the tested feature to the client, regression testing was made. Claim Outpayment Manager strongly relies on its' test automation to detect possible regression issues. Most of the OPM functionality is covered with over 500 automated UI tests. Test cases are created with selenium and robot framework to ensure that all of the functionality is working correctly and life cycles are followed.

As a result of current thesis, more than 100 of OPM previously used test cases were updated with new file processors. Test cases that use foreign outpayments now have to be handled with new outpayment processor.

During thesis work and implementation of SEPA feature there were no regressive issues found. Special care was taken with the old file format test automation. Tests were updated to verify that the parameters changes have no negative effect on normal payment export using domestic and foreign Bankgirot file formats.

4.6 Testing results

During testing phase the author of this thesis created and executed 78 defensive test cases, eight issues were found and five of those issues were corrected and retested. Additionally, most of the manual tests have been automated by the author. The main focus of automated test cases is the incoming and outgoing file data and formats. IBAN and BIC validations and concurrent usage of the system are not covered by the automated test cases.

As an example, Figure 13 presents the automated test results for domestic payment export.

Figure 13. SEPA payment export for domestic payments

Risks of file formats, parameterized implementation, and IBAN and BIC formats were addressed in the testing.

IBAN and BIC code formats are validated in the system. IBAN is verified by the country code, length of BBAN code, and control key. BIC code can only be validated against its length.

All parametrized possibilities were confirmed as a result of this thesis and each tested XML file format was accepted by SEB Test Bench service.

5 Summary

SEPA implementation is still very actual topic in Sweden and other non-euro countries. Swedish regulation also allows non-euro payments to be sent using the same format which allows to extend the possibilities originally intended by the SEPA regulation.

The aim of the bachelor's thesis was to produce an error prone implementation of the SEPA regulation with minimal risks for the customer.

As a result, identification of the modifications was listed down. In the scope of current thesis identified adjustments were covered by the quantitative risk analysis.

Identified risks from quantitative analysis included:

- XML file formats
- IBAN and BIC code updates
- parameterized implementation of payment export

These potential threats were mitigated and quality assurance was guaranteed by the author using exhaustive defensive testing. Defensive testing method was decided based on the risk analysis and work through of different testing techniques.

During this thesis work, author created a defensive test plan with 78 test cases and executed the test plan to ensure the quality of the product. Found issues were addressed by the development team and retested by the thesis author. Additionally, author of this thesis updated more than 100 of OPM previously used test cases with new file processors. Manual and automated test cases created and updated within this thesis are now used for future testing of the OPM module.

Analysed and tested solution was delivered to the Swedish life insurance company and is now successfully in production. This solution allows to create payments and read bank payment return information using ISO20022 format. All the biggest risks were identified and handled. Profit Software Ltd has received no issues from the client regarding the implementation.

References

[1] ISO 20022 Registration Authority, "The ISO 20022 Standard," [Online]. Available: https://www.iso20022.org/the_iso20022_standard.page. [Accessed 15 05 2017].

[2] Profit Software Ltd, "Profit Life&Pension, PLP Claim Outpayment modules and functionalities," [Online]. Available: http://www.profitsoftware.com/solution/plp-claim-outpayment-manager/. [Accessed 14 05 2017].

[3] Symantec Corporation, "Encryption, powered by PGP," [Online]. Available: https://www.symantec.com/products/information-protection/encryption. [Accessed 20 05 2017].

[4] Profit Software Ltd, "Profit Life&Pension," [Online]. Available: http://www.profitsoftware.com/solution/profit-lifepension/. [Accessed 14 05 2017].

[5] Microsoft, "Regular Expression Language - Quick Reference," [Online]. Available: https://msdn.microsoft.com/en-us/library/az24scfc. [Accessed 15 05 2017].

[6] Robot Framework Foundation, "Robot Framework, Generic test automation framework for acceptance testing and ATDD," [Online]. Available: http://robotframework.org/#introduction. [Accessed 15 05 2017].

[7] Skandinaviska Enskilda Banken AB, "SEB Test Bench," [Online]. Available: https://sebgroup.com/large-corporates-and-institutions/our-services/transaction-services/for-corporations/cash-management/integration-services/test-bench. [Accessed 15 05 2017].

[8] "REGULATION (EU) No 260/2012 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL," [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32012R0260&from=EN. [Accessed 15 05 2017].

[9] SPI Inc., "Sisu documents - structuring, publishing in multiple formats and search," [Online]. Available: https://packages.debian.org/jessie/text/sisu. [Accessed 15 05 2017].

[10] W3C, "Extensible Markup Language (XML)," [Online]. Available: https://www.w3.org/XML/. [Accessed 15 05 2017].

[11] European Central Bank, "The Single Payment Area (SEPA)," [Online]. Available: http://www.ecb.europa.eu/paym/retpaym/shared/pdf/SEPA_e-brochure_2013.pdf?ff1cd1d4060ec1655bcbf933fd3cbc97. [Accessed 12 05 2017].

[12] Skandinaviska Enskilda Banken AB, "SEPA and non-euro countries 2016 - and overview," [Online]. Available: http://sebgroup.com/large-corporates-and-institutions/our-services/transaction-services/sepa/sepa-and-non-euro-countries-2016-and-overview. [Accessed 15 05 2017].

[13] European Payments Council, "ISO 20022 Message Standards," [Online]. Available: http://www.europeanpaymentscouncil.eu/index.cfm/sepa-credit-transfer/iso-20022-message-standards/. [Accessed 15 05 2017].

[14] GV Solyushans Ltd, "List of abbreviations used in bank funds transfers," [Online]. Available: https://www.iban.com/whatis.html. [Accessed 20 05 2017].

[15] Registration Authority for ISO13616, "IBAN registry," [Online]. Available: https://www.swift.com/sites/default/files/resources/swift_standards_ibanregistry.pdf. [Accessed 15 05 2017].

[16] SEPA for Corporates, "IBAN Number Format by SEPA Country," [Online]. Available: http://www.sepaforcorporates.com/single-euro-payments-area/iban-number-format-sepa-country/. [Accessed 19 05 2017].

[17] Danske Bank Group, "What is IBAN, SWIFT and BIC," [Online]. Available: https://www.danskebank.ie/en-ie/personal/day-to-day/payments/helpfulinformation/pages/what-is-iban-swift-bic.aspx. [Accessed 13 05 2017].

[18] TransferWise Ltd, "SWIFT / BIC code toolkit," [Online]. Available: https://transferwise.com/swift-codes/. [Accessed 15 05 2017].

[19] Skandinaviska Enskilda Banken AB, "IBAN - International Bank Account Number," [Online]. Available: http://sebgroup.com/large-corporates-and-institutions/prospectuses-and-downloads/iban. [Accessed 13 05 2017].

[20] European Central Bank, "Glossary of term related to payment, clearing and settlement systems," [Online]. Available: https://www.ecb.europa.eu/pub/pdf/other/glossaryrelatedtopaymentclearingandsettlementsystemsen.pdf. [Accessed 15 05 2017].

[21] "REGULATION (EU) No 260/2012 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL, establishing technical and business requirements for credit transfers and direct debits in euro and amending Regulation (EC) No 924/2009," [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02012R0260-20140131&from=EN. [Accessed 14 05 2017].

[22] Techopedia Inc., "Risk Analysis," [Online]. Available: https://www.techopedia.com/definition/16522/risk-analysis. [Accessed 15 05 2017].

[23] Software Testing Fundamentals, "Software Quality Assurance," [Online]. Available: http://softwaretestingfundamentals.com/software-quality-assurance/. [Accessed 10 05 2017].

[24] D. Graham, E. v. Veenendaal, I. Evans and R. Black, "ISTQB EXAM CERTIFICATION, FOUNDATIONS OF SOFTWARE TESTING," [Online]. Available: https://www.utcluj.ro/media/page_document/78/Foundations%20of%20software%20testing%20-%20ISTQB%20Certification.pdf. [Accessed 20 05 2017].

[25] ISTQB Exam Certification, "What are the principles of testing?," [Online]. Available: http://istqbexamcertification.com/what-are-the-principles-of-testing/. [Accessed 19 05 2017].

[26] ReQtest, "What is boundary value analysis and quivalence partitioning?," [Online]. Available: http://reqtest.com/testing-blog/what-is-boundary-value-analysis-and-equivalence-partitioning/. [Accessed 15 05 2017].

[27] SOFTWARE TESTING HELP, "What is Negative Testing and How to Write Negative Test Cases?," [Online]. Available: http://www.softwaretestinghelp.com/what-is-negative-testing/. [Accessed 15 05 2017].

[28] S. Dorman, "What is “Defensive Programming”?," [Online]. Available: http://scottdorman.github.io/2008/07/04/what-is-ldquodefensive-programmingrdquo/. [Accessed 20 05 2017].

[29] S. Miller, "Top 10 Negative Test Cases," Pragmatic Software Co., Inc., [Online]. Available: https://www.techwell.com/sites/default/files/articles/XUS29743430file1_0.pdf. [Accessed 15 05 2017].

[30] Gurock Software GMBH, "Software Test Automation Tools," [Online]. Available: http://www.testingtools.com/test-automation/. [Accessed 15 05 2017].

[31] Selenium Project, "Test Automation for Web Applications," [Online]. Available: http://www.seleniumhq.org/docs/01_introducing_selenium.jsp#introducing-selenium. [Accessed 21 05 2017].

[32] J. Williams and R. Wandhöfer, "Minimising SEPA Migration Risk: A Practical Approach," [Online]. Available: https://www.treasury-management.com/article/1/151/1326/minimising-sepa-migration-risk-a-practical-approach.html. [Accessed 15 05 2017].