Esprit project 29512

Deliverable
I3.1.2
-
Method Adaptation Notes 2

Identifier	I3.1.2
Type	Deliverable
Activity	T3.1
Date	May 1999
Status	Version 1
Responsible partner	Lancaster
Availability	All project partners

Executive Summary

The move to European Economic and Monetary Union (EMU) began on 1st January 1999 when conversion rates of the participating member states were locked and the Euro (ε) introduced. This also signalled the start of a transition period, which will end in June 2002, during which trading can be in either national currency units (NCU) or ε. ε notes and coins, however, will not be introduced until 1st January 2002 at which time national currencies will be withdrawn from circulation. This means there is a three-year period when ε trading is by electronic means only.

EMU means that those software systems which process financial information in any of the participating member states must, eventually, be converted from national currency units to εs. As a result, many technical and more general business problems will be experienced by organisations within the participating member states.

When considering evolving its IT systems to be ε compliant, a company must consider several solutions concerning the conversion. Any of these solutions may affect systems from different business functions differently. For example, if an organisation requires dual pricing, this will affect invoicing and ticketing system types but not General Ledger accounting system types. It will undoubtedly mean that such things as package sizes can no longer be expressed in convenient money values for example £1.99, £24.95. This means that the solution for EMU facing an organisation is not straightforward and must be considered carefully.

The εmergency project provides a method for assessing the effort involved in ε conversion. This will enable an organisation to prepare its business for the possible disruption over the conversion period and also to alert it to the costs and timing pitfalls of conversion. An estimate for EMU effort, enables an organisation to have a good indication of whether or not additional resources will be necessary to complete the work and allows it to plan for these.

The εmergency method devolves from the Renaissance method which assisted an organisation to assess its IT base to decide what type of reengineering it required to handle its legacy dilemma.

This document contains a description of the assessment and estimating process and the documents devised to support it.

Section 1 gives some background to EMU and the εmergency project.

Section 2 discusses considerations and constraints applicable to ε conversion.

Section 3 explains some of terms used in this document.

Sections 4 and 5 introduce the estimating documents devised, describe the relationships amongst them and their contents.

Sections 6 and 7 describe how these documents should be used to create an estimate of effort for ε conversion.

Appendix A describes a set of solutions which can be applied to implement ε conversion. Appendix B and Appendix C describe how the Renaissance method was tailored to arrive at the εmergency method for ε conversion. Appendix D details the documents which were devised for the method and Appendix E gives a brief overview of the Renaissance Method itself.

Table of contents

1. Background

The move to European Economic and Monetary Union (EMU) began on 1st January 1999 when conversion rates of the participating member states were locked and the Euro (ε) introduced. This also signalled the start of a transition period, which will end in June 2002, during which trading can be in either national currency units (NCUs) or εs. ε notes and coins, however, will not be introduced until 1st January 2002 at which time national currencies will be withdrawn from circulation. This means there is a three-year period when ε trading is by electronic means only. As a result of this timetable, many technical problems will be experienced by organisations within the participating member states as they move to comply with EMU regulations. Examples of such problems are:

- *The rounding problem*: This will apply during the transition period where trading is possible in either national currency units or in εs. Strict rules for currency conversion have been formulated for EMU but rounding anomalies in calculations are possible as a consequence of applying them. This can be problematic, for example, where reconciliation of accounts is required. The conversions of sums of money from NCUs to εs may cause imbalances in 'total' figures. The notion of 'equality', in these cases, may have to be redefined to tolerate small differences.
- *Dual reporting requirements*: During the transition period, there may be a requirement for financial reporting in both εs and NCUs. This will require the redesign of visual displays such as screens or reports and could cause major adjustments to the internal data structures and processing of computer programs.
- *Historical data conversion*: Many systems rely on historical financial data. This data will be expressed in NCUs. Once the transition to the ε has been made, there must be mechanisms to use this data alongside ε data. Again, this may involve adjustments to the data structures and processing of computer programs.

While the above technical problems will apply to all participating member states, there are others which are unique to one country or group of countries. For example:

- The current Italian currency does not use decimals so that existing financial systems may not be designed to use decimal arithmetic. In this case, major amendments are required in any area where monetary calculations are performed.
- In the UK and Ireland, 1 ε will be worth less than 1 £. Consequently, when a sum in pounds is converted to εs, the resulting number of digits in the ε sum may be increased. This is problematic in situations where there is a fixed amount of space allocated to monetary amounts; high order digits may be lost or may overflow.

Apart from these technical issues there are many more general business issues that will confront an organisation facing EMU. For example, many companies choose package sizes so that their products are priced at a specific level (e.g. in the UK, £1.99). EMU may mean that package sizes must be changed.

EMU means that those software systems which process financial information in any of the participating member states must, eventually, be converted from NCUs to εs. To give an idea of the financial dimension of the problem, analysts from the OVUM and Gartner Group estimate the overall cost of ε introduction will be between 200 and 350 billion ECUs. The adaptation of software systems will account for a large percentage of this. It can be seen that the introduction of the ε will create a budgetary nightmare for the wide majority of European IT managers, especially as it coincides with the so-called Y2K problem.

When considering evolving its IT systems to be ε compliant, a company must consider strategies covering several aspects concerning the conversion –

- Dual Pricing Display
- Type of ε conversion (mono>multi etc)

- Timing of conversion
- etc need more examples – see soln/mod table

The collection of strategies adopted by the company can be thought of as its Corporate ε Evolution Strategy (εvolution Strategy) and IT applications will need modifying as a consequence. The Corporate εvolution Strategy may affect the solutions for applications from different business functions differently. Hence the modifications to one application may be different from those to another application. For example, if, as part of its Corporate εvolution Strategy, an organisation requires dual pricing, this will affect the solutions for invoicing and ticketing application types but not General Ledger accounting application types.

In order to ensure a smooth IT transition to the ε, an organisation must understand how its computer applications, of which many may involve legacy systems, support its business functions. This will involve mapping out its main business functions and marking the corresponding IT modules on it. This process will enable a 'portfolio' applications to be created so that an understanding of the magnitude of the changes to each can be assessed and hence enable the organisation to prioritise the changes according to local circumstances and draw up a plan for implementing them. Supporting the process of selecting εvolution Strategies is the objective of the εmergency project.

The technical objectives of εmergency are:

- To assess the feasibility of applying the Renaissance Method, particularly those parts of the method concerned with evolution planning, to the problems of ε conversion
- To suggest method adaptations which will tailor it to this task

The Renaissance project devised a method to assist an organisation assess its IT base to decide what type of reengineering it needed to handle its 'legacy dilemma'. It helped to devise an evolution strategy for a system, and then provided guidance on how to plan and execute a controlled evolution project.

In the εmergency project we are not 'devising an evolution strategy' as envisaged in the Renaissance context. We are merely determining the most cost-effective way of transforming our current financial systems so that they can handle EMU. An organisation has to juggle the cost effectiveness of an εvolution Strategy with its impact on the business:

- will displaying dual pricing in a particular way give it a market lead over a competitor
- will converting its General Ledger systems right now and leaving the ticketing/pricing type systems until later make the overall transition easier to manage
- and so on - need more examples

So, in εmergency, we are not assessing a system in order to present higher management with a viable, cost-effective system evolution strategy. We are trying to assist them to choose the best corporate εvolution Strategy by estimating effort for conversion and so help them plan for EMU. This will enable them to prepare the business for the possible disruption over the conversion period and also to alert them to the costs and timing pitfalls of conversion. What a user of εmergency needs, is to be able to define possibly several corporate εvolution Strategies and ensure that the optimum one is chosen.

This document covers an overview of how the Renaissance method can be used to assist a company with this ε conversion problem. It contains a description of the assessment and estimating process and the documents devised to support it.

Section 1 gives some background to EMU and the εmergency project.

Section 2 discusses considerations and constraints applicable to ε conversion.

Section 3 explains some of terms used in this document.

Sections 4 and 5 introduce the estimating documents devised, describe the relationships amongst them and their contents.

Sections 6 and 7 describe how these documents should be used to create an estimate of effort for ε conversion.

Appendix A describes a set of solutions which can be applied to implement ε conversion. Appendix B and Appendix C describe how the Renaissance method was tailored to arrive at the εmergency method for ε conversion. Appendix D details the documents which were devised for the method and Appendix E gives a brief overview of the Renaissance Method itself.

2. Considerations/Constraints for ε

As stated in section 1, the main area of Renaissance with which εmergency is concerned is Evolution Planning. This involves assessment of the current situation of a system, assessment of the target situation and estimating the possible routes between the two to choose both the optimum target and effective route.

εmergency is corporation rather than system oriented. The target assessment can be considered to be a strategical prioritisation of applications for ε conversion depending on the εvolution Strategy the company requires. Management, who will have considered the market and company constraints prior to considering the technicalities, will have devised the εvolution Strategy and will want to know the technical implications of applying it. This is subtly different from the aims of Renaissance and can be achieved by 'feeding' the εvolution Strategy into the method as a project constraint.

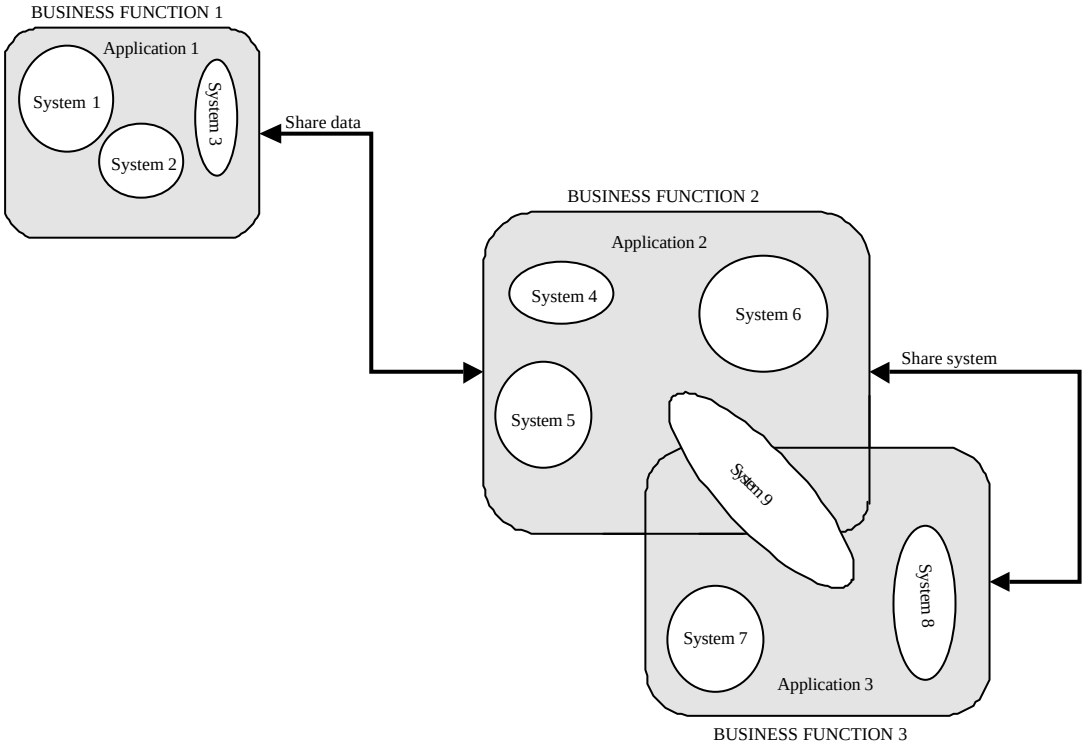


Figure 1 – Business functions and corresponding IT applications

It is the information gathering and estimating phases of Renaissance which are of concern to a user of εmergency. It is important to identify company and application characteristics which will assist ε conversion estimating remembering that a good project plan depends upon a good estimate of effort, costs and risks.

The scope of Renaissance is one system supporting a part of an organisation's business process. It supports evolution strategy formulation for that system in an incremental manner. As such, it uses System, Component (ie program) and Constituent levels for assessment purposes.

εmergency is 'organisation' oriented. That is, its scope covers all systems supporting an organisation's business process. It is important, therefore, to look at an organisation's Business functions.

To satisfy these 'applications' of IT are necessary. An application may consist of several systems as shown in Figure 1.

Bearing this in mind, for εmergency, 2 assessment levels have been defined:

- A *Corporate Level* - this is a preliminary, high level assessment pitched at the customer business level. This is a higher but narrower level that the Renaissance system level of assessment. It provides a way of identifying applications, estimating an initial rough order of magnitude of the effort of ε conversion and serves to drive further assessment
- *Application Level* – this is a lower level assessment than the Corporate Level and will be used for each application (see Figure 1). The accumulation of all application level assessments will enable an estimate of ε conversion effort at a sufficient level of detail to enable planning to be performed

The εmergency approach will be to do one Corporate Assessment to identify the necessary applications and calibrate the method and an Application Assessment for each application identified.

In accordance with the Renaissance policy of in use evaluation, the εmergency method has been devised to enable feedback into the estimating method from every application conversion as shown in Figure 2. This has been achieved by creating effort parameters and weightings which can be specified and applied independently from the actual application assessment. This has two advantages:

- It allows fine-tuning of the method in the light of actual effort figures
- It allows the method to be calibrated for any organisation

As the estimate is completed for each application, estimating experience can be fed back into the assessment process. As the corresponding conversion is completed, the actual effort involved can be recorded, via in use evaluation, enabling parameters and weightings to be perfected.

This enables:

- subsequent assessments to be estimated more accurately
- the conversion process itself to be streamlined

An organisation must decide what to do about the cost of converting data stored in files and databases. Applications are often linked by files therefore file conversion overhead is not necessarily application dependent. This is particularly true of databases where items such as the data dictionary, DDL and database regeneration overheads must be considered. 'Running' the method separately for files and databases represents a large overhead in time and effort. The εmergency approach to file conversion, therefore, is to add an overhead 'factor' into the estimate using the effort parameters and weightings mentioned above. For example, the factor will be high for a financial application utilising database technology, lower for a financial application not utilising database technology and non-existent, possibly, for a delivery routing application.

Having obtained an estimate for effort, an organisation will have a good indication of whether or not additional resources will be necessary to complete the work and be able to plan for these.

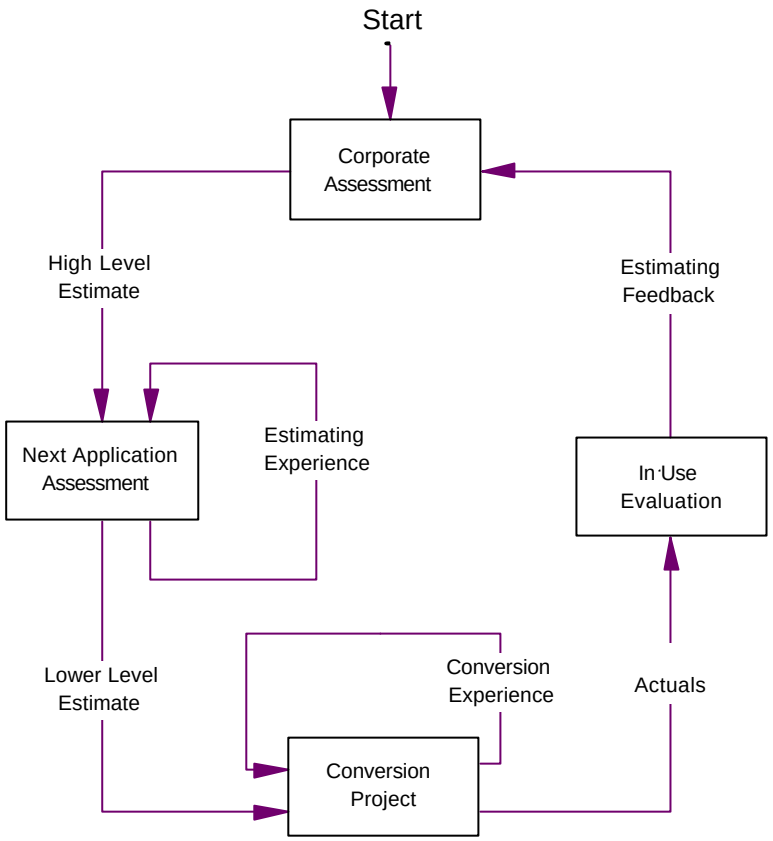


Figure 2 – The emergency Assessment and Conversion Cycle

There are two scenarios with respect to assessment for ε conversion as follows:

- An organisation may run the method for its own purposes to assist its ε conversion budgeting and planning activities
- A contractor may run the method in order to set up a contract with an organisation for performing ε conversion for it

The remaining sub sections in this section outline what is involved in the two assessment levels.

2.1 Corporate Assessment

This assessment records information necessary to the subsequent more detailed Application Assessments. It will involve creating a diagram of all business functions similar to that shown in Figure 1 and identifying the applications used by a company along with information which will allow the Application Assessment to proceed.

This assessment provides a way of:

- obtaining a quick order of magnitude of conversion effort for each application and hence a ball park figure for total ε conversion effort
- organising and planning the more detailed estimating tasks
- enabling applications to be prioritised for conversion
- enabling an organisation to assess its IT portfolio.
  For example an organisation may actually throw away a system and replace it in the light of

the assessment results. Alternatively, it may want to reengineer a system to a client/server platform and use ε conversion as an excuse to do it 'now'.

Collecting application information, at this level can begin before an εvolution Strategy has been fully formulated ie the two activities can be run in parallel.

The Corporate Assessment will involve:

- Defining the εvolution Strategy as a constraint to be considered during assessment
- Tailoring the method:
  - assigning staff to document application information
  - calibrating the method parameters to the organisation involved in the assessment (or re-calibrating it as part of in use evaluation)
- Creating an inventory of all applications[1] supporting the organisation and for each:
  - Listing the contacts responsible for the application
  - Determining the business area (Financial (and type ie GL, Tk/Pr), Non Financial)
  - Determining the original development of the application (In house, package etc) and the platforms it utilises
  - Using 'expert opinion' to determine the order of magnitude of the task ahead assigning each application
    - a size parameter[2]
    - a complexity factor

For details of tailoring the method at this level please see Appendix B.

2.2 Application Assessment

This provides a more detailed way of estimating the effort required to convert one application to be ε compliant. It enables:

- contractual details and procedures to be clarified
- staffing levels to be determined for the actual conversion
- infrastructure requirements to be formulated
- project planning activities to commence

Each application identified in the Corporate Assessment will be assessed. As system evolution requirements are not to be considered and a well defined conversion task is defined, this level will be narrower than envisaged in Renaissance. Essentially, we need to determine the number of money fields used, where they are used and how they are used in order to quantify the work content of changing them. This assessment will make use of the information recorded during the Corporate Assessment.

The Application Assessment will involve:

- Eliciting information concerning the application from the customer contacts identified in the Corporate Assessment
- Tabulating the information obtained
- Computing the conversion effort based on the information obtained
- Documenting the conversion effort

For details of tailoring the method at this level see Appendix C .

[1] For Siceas, this will obviously involve only the one system

[2] Must determine an algorithm for a quick and dirty estimate of size.
Mixture of informal expert knowledge and assessment information.

3. Terms used in this Document

3.1 Application / System

ε conversion (and hence, εmergency) is 'organisation' oriented. That is, its scope covers all applications supporting an organisation's business. It is important, therefore, to look at an organisation's business functions when assessing ε conversion work. To satisfy a business function, an 'application' of IT is necessary. An application may consist of several systems, as shown in Figure 1, and may interact with other applications.

In Figure 1 the three business functions would appear to be serviced by 3 applications. Applications 1 and 3 consist of 3 systems and application 2 of 4 systems.

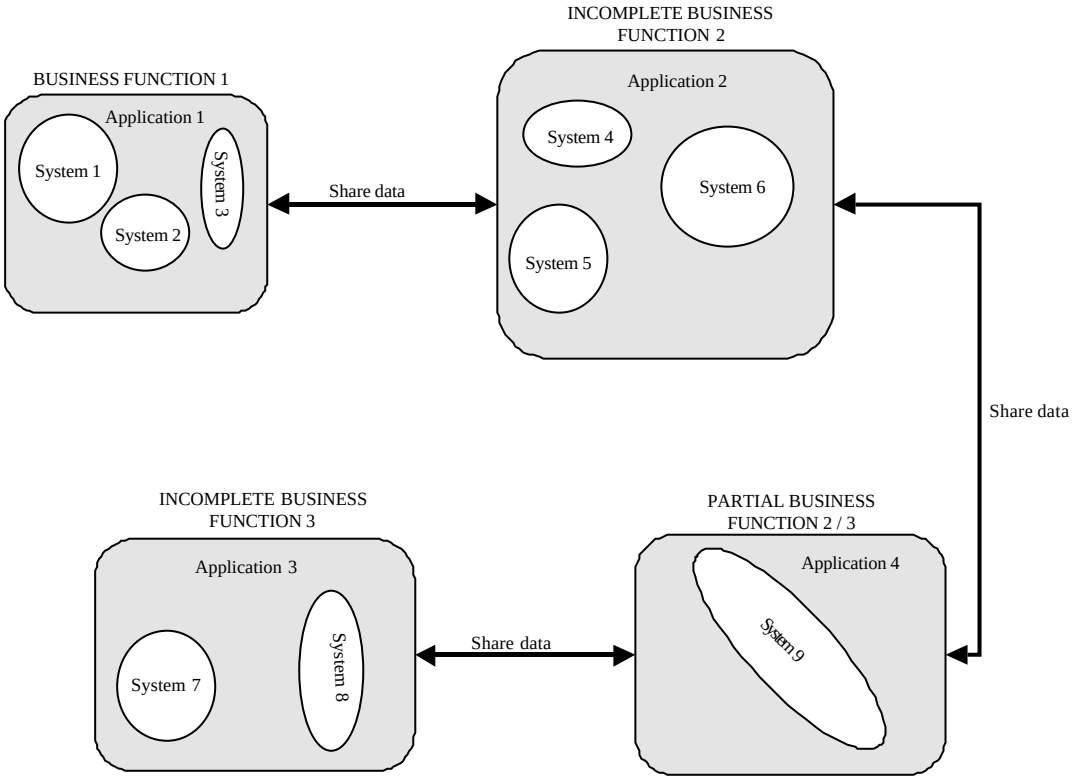


Figure 3 – Revised business functions with corresponding IT applications

However, applications 2 and 3 share one or more programs which belong to system 9. To avoid duplication and errors in estimating figures during the process of ε assessment, system 9 would be removed from both application 2 and application 3 and would be assessed as if it were an application in its own right. This is demonstrated in Figure 3**.**

In this case application 1 consists of systems 1, 2 and 3 each of which is wholly involved in satisfying business function 1. Application 2 consists of systems 4, 5 and 6 each of which is wholly involved in satisfying business function 2. Application 3 consists of systems 7 and 8 each of which is wholly involved in satisfying business function 2. Application 4 consists of system 9 which is wholly involved in satisfying business functions 2 and 3.

An **application** is a collection of systems each of which wholly services one or more business functions such that no program it contains can be counted twice. *Need a better way of saying this*

3.2 Customer / Assessor

A **customer** is the person or group of persons who owns or is responsible for an application as defined above.

An **Assessor** is the person or group of persons preparing the estimate. The assessor may, or may not, be from the same organisation as the customer.

3.3 Archives

For the purpose of ε assessment we consider the term **archive** to mean any collection of data. This may be in electronic, paper or fiche form. Electronic archives may be simple (flat files) or complex (VSAM or database tables etc).

3.4 Application Criticality

To prioritise an application for ε conversion we need to be able to assess its size, complexity and importance to the organisation performing the ε conversion. This combination of size , complexity and importance is termed application **criticality.**

3.5 The 7 Euro Solutions

In section 1 we said that when considering evolving its IT systems to be ε compliant, a company must consider strategies covering the aspects ε conversion and that IT applications will need modifying as a consequence. It is possible to identify several general solutions to satisfy these strategies. The εmergency team has identified 7 solutions which should cover the vast majority of requirements and which are described briefly in this section. Appendix A gives a more detailed outline of them along with a short guide to their evaluation and selection.

3.5.1 Multi Currency

This is a solution which involves adapting an application to handle any currency including ε, NCU and others such as Yen and Dollar. While the application produced is highly flexible, the solution is complex and requires a large investment in effort and cost.

3.5.2 Duplicate Application

This solution involves duplicating the application, converting one to handle the ε and managing both during the transition period. While the converted application will be usable at the end of the transition period, there is the overhead of managing two applications during it to be considered.

3.5.3 Dual Currency

This solution involves the ability of the application to input, output and store monetary amounts in either ε or NCU with a code to distinguish which. However, wherever possible, outputs will remain in NCUs.

This is a simplified version of the multi currency solution and, providing it is in place within 1999, allows progressive introduction of the ε.

3.5.4 Dual Amounts

This solution involves the ability of the application to input and output monetary amounts in either ε or NCU, as for Dual Currency above, but storing both the amounts. This means that input is always duplicated and will need handling consistently. Again, providing it is in place within 1999, it allows progressive introduction of the ε.

3.5.5 Masking National Currency

This solution requires modifying an application to allow for direct input of NCUs but input of εs through the use of a converter. Storage is in NCUs and output for ε, if required is via a converter. This is a temporary solution for the transition period and is applicable only when the number of monetary amounts being handled is small.

3.5.6 Masking ε

This is similar to the previous solution except storage is in εs and output in NCUs through the use of a converter. This solution requires the modification of an application to handle ε before the commencement of the transition period however at the end of the transition period, it is a simple matter to remove the converter.

3.5.7 Single Currency ε

This solution requires the modification of an application operating in NCUs to operate in ε only. While this creates an application which will be usable after the transition period, it requires further actions to be taken to cope with the transition period itself.

3.6 Application Modifications

The 7 solutions outlined above generate application modifications. A table of which modifications are relevant to any solution has been devised by the εmergency team and documented in section 5.6. A sample can be seen in Appendix D. The relationships amongst the solutions and modifications are actually far more complex than shown in the table.

For example, if modifications for decimalisation are being made, this may require archives to be modified. If an archive is to be modified, then this will affect both the format (conversion of programs and user interfaces – software modifications) and actual data (conversion of archive data – hardware modifications). The complexity of archive conversion also depends upon the type of archive itself (see section 3.3). If the archive in question is used by another application which is not scheduled for conversion until later then a converter of some sort must be created to enable the two applications to communicate over the conversion period.

There is no single recipe for determining the complete set of modifications for any one application. It depends upon the application and its individual environment. The detail must be identified, however, before an estimate for ε conversion can be made with any real confidence.

3.7 Cost Items

For speed we need to be able to estimate ε conversion effort from the solutions and modifications described above using aspects of IT which are readily available and which enable us to judge an application's criticality in a short space of time. In general, the following items can be counted relatively easily:

- TP programs
  These items represent the processing 'behind' the interactive user interfaces
- Batch programs
  These items represent the off line processing
- Maps
  These items represent the interactive side of the user interface. They are the display maps which may have additional processing associated with them
- Archives
  These are data items whose formats may need modifications

- Interfaces
  An interface is a program needed to link one application with another. The total number of interfaces for an application depends upon the I/O flows between it and other applications
- Prints
  These items represent the hard copy side of the user interface. They are the reports and information printed on any customised or preprinted forms

The items above are IT items which may include monetary amounts and can be counted in a reasonably short time span. As most organisations will have performed a Y2K exercise, an inventory of these items probably exists. Together with the converters mentioned in the previous sub sections, they represent **cost items** upon which an estimate can be built.

3.8 *e* Conversion Dependencies

Having obtained a count of cost items, it is possible to multiply them by effort parameters (in person/days) to arrive at an estimate of effort for ε conversion. For example, if an organisation establishes, from historical management data, that it takes an average of x person days to modify a batch program, then the total effort to modify n batch programs is n * x.

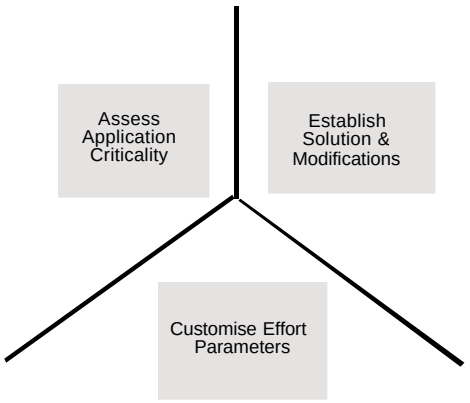


Figure 4 – Establishing Effort

While the effort depends, essentially, upon the cost items above, conditions of additional complexity due to the types of modifications chosen must also be considered. This means that an estimate for ε conversion depends upon 3 factors:

- The application criticality
- The solution and modifications required
- The calibre of the personnel performing the conversion tasks

As such there are 3 major tasks to be performed to create an estimate of effort as shown in Figure 4

Finally, contingency must be added to cover any inadequacies in supplied information, competence of the customer or size and complexity of the organisation itself.

3.9 Establishing Effort Parameters

Just as organisations themselves differ widely, the capabilities of the personnel performing ε conversion will also differ widely from organisation to organisation. This means that a set of effort parameters is required which can be defined for any organisation. Effort parameters must be laid out in the form of tables to which an assessor can refer. Bearing in mind the information in the previous sections, the following effort parameter types must be devised:

- To determine the proportion of cost items which will be likely to be involved in ε conversion
- To enable effort to be quantified from the cost items related to application criticality
- To enable additional effort to be quantified from conditions of added complexity due to the modifications required to service a ε solution
- To enable effort for contingency to be quantified based upon weightings which establish confidence in the figures supplied

3.10 Establishing Effort

To estimate effort using the effort parameters, εmergency splits the estimate for adaptation into two distinct areas:

- *Implementation Effort*
  This consists of the direct effort to perform the adaptation of the application for ε conversion without consideration for information accuracy and consists of:
  - The *Basic Effort* depending purely upon the size of the conversion. That is, on using the number of programs (TP and Batch) and maps, within an application, which need modification
  - The *Extra Effort* depending upon the complexity of the conversion or the peculiarities of the application in question. That is, on using the type of ε solution chosen and the resulting modifications. Extra effort will usually come through archives, interfaces, prints, algorithms embedded in the programs and any additional functionality required
- *Other Effort* depending upon the accuracy of the information obtained. That is competence and reliability of the customer, the size and complexity of the organisation and accuracy of information supplied

4. Emergency Assessment and Estimating Documents

The creation of an estimate for ε conversion will require a set of documents which will cover the information described in the previous sections. This section identifies and describes these documents and demonstrates the relationships between them. The following categories of documents are required:

0. A document to catalogue identified applications
1. A document the customer can use to capture information concerning an application
2. Documents tabling weightings the assessor can use to asses the customer and the information this customer supplies
3. Document to summarise application criticality
4. Document to table the various ε solutions and the application modifications each will require
5. Document to summarise the modifications required to apply the chosen solution to an application
6. Documents to table efforts to be applied to cost items
7. Document to record the final estimate

4.1 Table of documents

The table below gives the names of the documents used to prepare a ε estimate.

Id	Category	Name
DOC0	0	Application Catalogue
DOC1	1	Customer Questionnaire

DOC2	2	Assessor Questionnaire
DOC3	2	Direct Effort Weights
DOC4	3	IT Characteristics
DOC5	4	Modification/Solution Table
DOC6	5	Euro Solution Characteristics
DOC7	6	Cost Item / Modification Parameters
DOC8	6	Effort Parameters
DOC9	6	Other Cost Parameters
DOC10	7	Estimate

Table 1 - Documents cross-reference

4.2 Relationships Amongst Documents

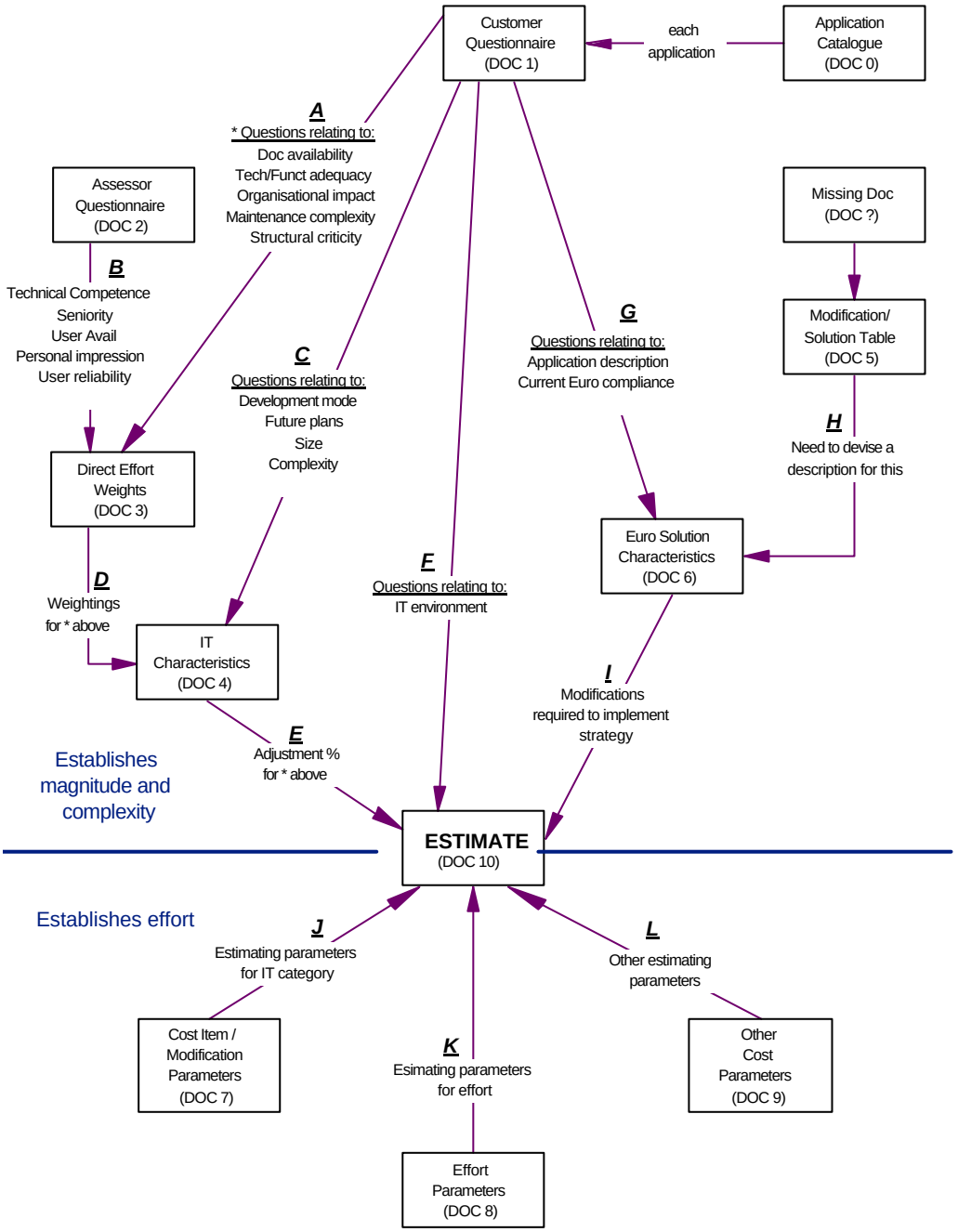


Figure 5 – Document Relationships

5. Document Descriptions

Why is it that some columns are missing from the various spreadsheet tables????

5.1 DOC0 - Application Catalogue

Completed by Assessor

Function To identify the applications which support a customer's business functions

Used in conjunction with	DOC1

This form lists all the applications involved in supporting a customer's business functions. For each application it will contain information which will be necessary to complete the Application Assessment. For example the list of personnel with specialised application knowledge.

5.2 DOC1 - Customer Questionnaire

Completed by	Customer
Function	To supply information about an application which will enable an estimate of effort to be made to make it ε compliant
Used in conjunction with	DOC3, DOC4, DOC6, DOC10

It consists of the following sections

0 Cover Page	
1 General data	questions 1.1 to 1.12
2 Package information	questions 2.1 to 2.10
3 Technical Aspects	questions 3.1 to 3.9
4 Functional Aspects	questions 4.1 to 4.11

Each of these forms relates to one application. This may be built up from several systems as previously explained in Figure 1 and Figure 3. It may be that people from many departments will hold the information needed to complete this form. In this case, an individual form will be completed by each and the information condensed into a 'final' one for estimating purposes. Each application may be implemented on up to 3 platforms (Host, C/S, PC) therefore it is possible for there to be as many as 3 of these 'final' forms for any one application.

5.3 DOC2 - Assessor Questionnaire

Completed by	Predefined as part of Corporate Assessment
Function	Gives information about the confidence the assessor may have in the information supplied by the customer on the Customer Questionnaire. It forms the basis of a risk exercise
Used in conjunction with	DOC3

The assessor uses this to determine 'user reliability'.

It results in an adjustment factor (low, medium, high) to be applied to effort weightings (DOC3) bearing in mind the customer's level of responsibility, technical competence, time spent in the role, availability and personal impression of the assessor.

This form will need calibrating to reflect the conditions within the organisation performing the ε conversion.

5.4 DOC3 - Direct Effort Weights

Completed by	Predefined as part of Corporate Assessment
Function	Documents weightings to be used to adjust the implementation effort taking into account the reliability of the IT characteristics used to assess

	an application.
Used in conjunction with	DOC1, DOC2, DOC4

This table is used to estimate the amount of contingency to be added to the final estimate because of poor information or customer quality. The assessor uses the customer's answers to questions on the Customer Questionnaire (DOC1) and the 'user reliability' weightings from the Assessor Questionnaire (DOC2) to determine low, medium, high rating for 5 ε conversion characteristics as follows:

1.8 to compute a Document Availability weighting.
1.7, 1.10, 1.11, 1.12 to compute a Technical / Functional Adequacy weighting.
4.1, 4.2, 4.6 to compute an Organisational Impact weighting.
1.4, 1.6, 3.1, 3.2, 3.8, 3.9 to compute a Maintenance Complexity weighting.
3.4, 3.7, 4.7, 4.8, 4.9 to compute a Structural Criticality weighting.

This form will need calibrating to reflect the conditions within the organisation performing ε conversion.

5.5 DOC4 - IT Characteristics

Completed by	Assessor
Function	To record details which enable the criticality of an application to be determined
Used in conjunction with	DOC1, DOC3, DOC10

The assessor uses the customer's answers to questions on the Customer Questionnaire (DOC1) and the weightings devised using DOC3 as follows:

Identification information		
COL A	Company	From DOC1 Cover page
COL B	Area	From DOC1 Cover page
COL C	Sub-system	From DOC1 Cover page
COL D	Application	From DOC1 Cover page
Information concerning the original development of the system.		
COL H	Devel.Policy	From DOC1 Question 1.3
COL I	App Type	From DOC1 Question 1.3
Information concerning the future of the system		
COL J	Repl. Before Euro	From DOC1 Question 1.6 (Fill with the date if the replacement is before EURO)
Information concerning the size of the system		
COL K	Pgm TP	From DOC1 Question 3.4
COL M	Pgm batch	From DOC1 Question 3.4
COL N	Total Pgm	Col_K+Col_M
COL O	Maps	From DOC1 Question 3.4
COL U	Tot arc	From DOC1 Question 3.7
COL V	Prosp/Module/P	From DOC1 Question 4.7

	rint	
Information concerning the confidence in the customer's figures		
COL W	tech/funct adeq	(W2) In conjunction with DOC3 Technical/Functional Adequacy (Low, Medium, High)
COL X	organi. Impact	(W3) In conjunction with DOC3 Organisational Impact (Low, Medium, High)
COL Y	structural criticality	(W4) In conjunction with DOC3 Structural Criticality (Low, Medium, High)
COL Z	mainten complexity	(W5) In conjunction with DOC3 Maintenance Complexity (Low, Medium, High)
COL A	Document level	(W1) In conjunction with DOC3 Documentation Availability (Low, Medium, High)
Information concerning the complexity of the system		
COL AB	Nr of interf	From DOC 1, sum of values in answer 4.8 and 4.9

Each line on the form represents a single application. Information for several applications may be recorded on this form. Each application may have up to 3 aspects (Host, C/S, PC) therefore it is possible for there to be as many as 3 lines on this form for any one application.

5.6 DOC5 - Modification / Solution Table

Completed by	Predefined by the εmergency team
Function	Relates euro solutions to system modifications
Used in conjunction with	DOC6

For any of the 7 ε solutions (see section 3.5) this table identifies application modifications needed. There are basically 11 types of modifications which can be made to an application as described in section 3.6:

- Decimalisation – M1
- Dual Currency Management– M2
- Currency Code Management – M3
- Computation Algorithm Changes – M4
- Modification of Constants and Representations – M5
- New Functions – M6
- Dual Currency Output – M7
- Interface Converters – M8
- Archive Converters – M9
- TP Input Converters – M10
- Print Converters – M11

For example, if a solution is for Dual Currency, then it shows that modifications are required for each entry in the column under Dual Currency which contains a yes. That is decimalisation, currency code management, change of algorithms and constants, new functions, interface file converters and three optional ones: output in dual amount, interface to other applications, print converters.

5.7 DOC6 - Euro Solution Characteristics

Completed by	Assessor
Function	To identify the modifications required to an application based upon the solution which has been chosen for it.
Used in conjunction with	DOC1, DOC5

The assessor uses the customer's answers to questions on the Customer Questionnaire (DOC1) as follows:

Identification information		
COL A	Appl Aronym	From DOC1 Cover page
COL B	Description	From DOC1 Question 1.1
Euro compliancy assessment		
COL C	Curr Code	From DOC1 Question 1.9
COL D	Decimals	From DOC1 Question 1.9
COL E	Multi-curr	From DOC1 Question 1.9
COL F	Appl Euro compl?	From DOC1 Question 1.9
COL I	Limit deployment date	If there is a limit date by when the application must be released
Solution		
COL J	Solution Type	(Sx) Type of solution to be applied for the adaptation of the application. Possible types are given in DOC5.
COL K	Impact	This is evaluated by the assessor and concerns the complexity of the adaptation (High, Medium, Low, None). The usage of None means that the adaptation is not required (e.g: the application is already Euro compliant) or not suggested (e.g. the application will be replaced before Euro introduction).
COL L	Notes	Supportive information for choice of Impact (if necessary)
Modifications Required		
COL M	Decimaliz	(M1) Depends upon the Solution Type. If required, the number of archives involved must be recorded.
COL N	Dual Amount	(M2) Depends upon the Solution Type.
COL O	Curr Code	(M3) Depends upon the Solution Type.
COL P	Algorit	(M4) Depends upon the Solution Type. If required, it is necessary to specify if algorithms provide special complexity
COL Q	Constants & Represent	(M5) Depends upon the Solution Type.
COL R	New Functions	(M6) Depends upon the Solution Type. If new functions are required, the number needs to be estimated and recorded
COL S	Dual Amount output	(M7) Depends upon the Solution Type. If dual amounts have to be output, the number of functions necessary must be estimated and recorded

COL T	Interfacing convert	(M8) Depends upon the Solution Type. If required, the related complexity level must be recorded
COL U	Arch convert	(M9) Depends upon the Solution Type. If required, it is the related complexity level must be recorded.
COL V	TP input convert	(M10) Depends upon the Solution Type. If required, the related complexity level must be recorded.
COL W	Prints Convert	(M11) Depends upon the Solution Type. If converters are required, the number of additional functions must be estimated and recorded
COL X	Other modifications	(M12) Depends upon the Solution Type. If converters are required, the number of additional functions must be estimated and recorded
Package Conversion Costs		
COL Y	Cost for package or service (annual fee)	Record contract amount (only if the application is a package of "outsourcing")
COL Z	Annual fee for package	Record maintenance contract amount (only if the application is a package)
COL AA	Cost for Euro adaptation by the provider	Record amount (only if the application is a package of "outsourcing")

The assessor will need to discuss with the customer the ε solution to be applied to this application then identify the relevant modifications using the Modification / Solution Table (DOC5). The results will be recorded in this table.

The assessor must find information about the application not recorded on the Customer Questionnaire to fill in these columns eg do any algorithms create additional complexity to the system.

Each line on the form represents the solution characteristics for an application. Information for several applications may be recorded on this form. Each application may have up to 3 aspects (Host, C/S, PC) therefore it is possible for there to be as many as 3 lines on this form for any one application.

5.8 DOC7 – Cost Item / Modification Parameters

Completed by	Predefined as part of the Corporate Assessment
Function	A model which provides the standard impact percentage of required modifications for each cost item.
Used in conjunction with	DOC10

Not all programs, maps, archives and interfaces for a given application will need to be modified for ε conversion. Experience has shown that only a certain percentage will need modification. This form shows the percentage which will be likely to require modifications for each of the 11 modification types. For example, if you must do modifications for decimalisation, algorithms and constants, the percentage of programs and maps which need modifying is given in DOC7 as:

- decimalisation (70% TP programs, Batch programs and Maps)
- algorithms (30% TP programs and 50% Batch programs)
- constants (50% TP programs, 30% Batch programs and 75% Maps)

The table is built such that the lower percentages are subsumed in the higher ones so the percentage to be chosen is NOT the sum of the percentages of all the modifications, but the percentage associated with the modification which has the largest. Hence the percentages to use in the example above are 70% for TP programs, 70% for Batch programs and 75% for Maps as these are the largest in each category.

5.9 DOC8 - Effort Parameters

Completed by	Predefined as part of the Corporate Assessment
Function	Contains effort parameters to be applied to cost items depending upon the modifications required
Used in conjunction with	DOC10

This form documents figures for implementation effort (basic effort and extra effort) in person days for the various platform types and complexity factors. From experience it has been discovered that it takes the same amount of effort to modify a TP program as a Batch program.

The basic effort figures cover

- Decimalisation (M1), Dual Currency (M2), Currency Code (M3), (simple) Algorithms (M4) and Constants (M5) modification types in section 1 of the table

The extra effort figures cover

- complex archives (M9) and algorithms (M4) in section 2 of the table
- converters (interface (M8), archives (M9), TP input (M10) and prints (M11)) in section 3 of the table
- new functions (M6 and M7) in section 4 of the table

This form will need calibrating to reflect the capabilities of the personnel assigned to ε conversion tasks. That is, it will reflect the technical competencies of the personnel for any given platform. These personnel may be from the organisation performing the conversion or outside contractors.

5.10 DOC9 - Other Cost Parameters

Completed by	Predefined as part of the Corporate Assessment
Function	Contains parameters to be applied to determine ‘other’ costs
Used in conjunction with	DOC10

This is the means of adding contingency into an estimate, in the form of other costs, for poor information or customer quality and organisation complexity. It assumes that other costs are determined as a percentage of implementation effort. Percentages to be added for contingency due to Structural Criticality, Maintenance Complexity, Organisational Impact, Documentation Availability, Technical / Functional Adequacy, are recorded on this form.

The resulting percentage for other effort is the sum of the relevant percentages for each weighting and should be limited to a fixed upper percentage of 40% and lower percentage of 5%. This means, in effect, that other costs can not be less than 5% or more than 40% of implementation costs.

This form will need calibrating to reflect the conditions within the organisation performing ε conversion.

5.11 DOC10 - Estimate

Completed by	Assessor
Function	To record the final estimate for ε conversion of an application
Used in conjunction with	DOC1, DOC4, DOC6, DOC7, DOC8, DOC9

This form documents the effort for ε conversion for applications within a customer's organisation.

Each line on the form represents the estimate for an application. Information for several applications may be recorded on this form. Each application may have up to 3 aspects (Host, C/S, PC) therefore it is possible for there to be as many as 3 lines on this form for any one application.

Identification information		
COL A	Area	From DOC1 Cover page
COL B	Application	From DOC1 Cover page
COL C	Acronym	From DOC1 Cover page
COL D	Platform	From DOC1 Question 3.2
COL E	Estimate	Value is 'No' if DOC6 COL K is 'none'. This implies that no estimate is to be done. Yes otherwise
Estimate		
COL F	Implement Effort (p/d)	Is the implementation effort for the Conversion. If a package, this effort is related to integration and customisation. See calculations in section 6.
COL G	Other Effort (p/d)	Other effort. See calculations in section 6.
COL H	Total Effort (p/d)	Total effort is the sum of COL F and COL G
COL I	Limit Date	DOC6 COL I if available.
COL J	Notes	Supporting information and notes.

6. Creating a ε Estimate

This section describes how to build up an estimate for ε conversion using the forms and method devised in the εmergency project. A description of how this method devolved from the Renaissance method can be found in Appendix B and Appendix C .

The effort for making an application ε compliant depends, basically, upon the number of programs and maps it contains which need modifying plus an additional overhead for other items which depend largely upon the ε solution chosen and the quality of the information supplied. So the method involves collecting and recording application and solution information using the estimating documents described previously.

6.1 Step 1 – Identify Applications

The method first directs the assessor to gather information about the corporate IT base of the customer. This represents the Corporate Assessment. The assessor liaises with the customer to identify applications for ε conversion as described in section 2.1.

If guestimates of size and complexity can be made for each application, then a very rough order of magnitude for ε conversion can be made from this catalogue.

Each application identified is recorded on DOC0, the Application Catalogue (see section 5.1).

This step is used to calibrate the weightings and effort parameters recorded in DOC2, DOC3, DOC7, DOC8 and DOC9 to reflect the calibre of the assessor, customer personnel and information supplied. The documents supplied in Appendix D show example calibrations.

If in use evaluation is being performed, then this step is used to recalibrate the method with known figures.

Once the application catalogue is complete, steps 2 to 5, representing, the Application Assessment are performed for each application it contains as described in section 2.2. The information in the catalogue allows a master plan for the Application Assessment to be created.

6.2 Step 2 - Gather Application Information

The method directs the assessor to gather information about the application and its operational environment and assess the quality of this.

6.2.1 Collect Application Information

The assessor requests the relevant customer contact to complete DOC1, the Customer Questionnaire (see section 5.2). Where an application utilises more than 1 platform type, one of these forms is filled out for each. That is, each platform type is considered to be a separate application.

6.2.2 Determine Quality of Application Information

The application information supplied may not be accurate. This may be because of the inefficiency of the customer or the quality of the information supplied. The assessor must quantify weightings which indicate the confidence in the information supplied by the customer. The assessor does this by:

- using DOC2, Assessor Questionnaire (see section 5.3) to determine a weighting for user reliability
- using DOC3, Direct Effort Weights (see section 5.4) in conjunction with DOC1 and the user reliability weighting above to determine weightings for:
  - Documentation Availability – W1
  - Technical/Functional Adequacy – W2
  - Organisational Impact – W3
  - Maintenance Complexity – W4
  - Size and Complexity – W5

6.2.3 Document Application Information

The information in DOC1 and the 5 weightings are used by the assessor to complete DOC4, IT Characteristics (see section 5.5), which gives information about the original development of the application, information concerning its future, and information concerning its size and complexity along with the weightings quantified above.

6.3 Step 3 - Gather ε Solution Information

The next step of the method is to gather information concerning the ε solution for the application in question and the type of modifications each will require.

6.3.1 Determine Solution

There are basically 7 solutions which can be applied to an application to satisfy the Corporate εvolution Strategy:

- Multiple Currency – S1
- Duplicate Application – S2
- Dual Currency – S3
- Dual Amounts – S4
- Mask National Currency – S5
- Mask Euro – S6
- Single Currency Euro – S7

Each of these is described in section 3.5. The assessor must discuss the type of solution with the customer in order to determine the modifications which the application will require. Once the solution is determined, it is noted on DOC6, Euro Solution Characteristics (see section 5.7).

6.3.2 Determine Modifications

The solution chosen by the customer will determine the types modifications required to the application as described in section 3.6. These modifications have been tabled in DOC5, the Modification/Solution Table (see section 5.6).

Using DOC5, the assessor determines the modifications to be made to the application programs based on the chosen solution.

6.3.3 Document Solution Modifications

The information in DOC1 and DOC5 is used by the assessor to complete DOC6, Euro Solution Characteristics (see section 5.7), which gives information about the original development of the application, a ε compliance assessment and indications as to what modifications are required. The assessor will complete the table by 'coding' the required modifications as follows:

6.3.3.1 Decimalisation – M1

This is used to determine the effort required to change the formats for the archives for decimalisation. That is, the software aspect of the conversion. This will apply only to those countries whose NCUs do not currently contain decimals.

0 – No decimalisation required

If decimalisation is required the assessor must ascertain whether archives are to be converted and, if so, the number to be converted.

1 – Decimalisation required but no conversion of archives

2 – Decimalisation required with conversion of archives.
In this case the assessor must also document the number of archives by coding a plus sign followed by the number

Archive conversion may be an issue for UK and Ireland whose amounts in ε will be larger than in NCUs??? Perhaps we should rename this so that it is more generic????

6.3.3.2 Dual Currency Management – M2

0 – not required

1 – required

6.3.3.3 Currency Code Management – M3

0 – not required

1 – required

6.3.3.4 Computation Algorithm Changes– M4

0 – not required

If changes are required to computation algorithms, the assessor must determine the complexity of them and state this here.

1 – required but algorithms are simple

2 – required but algorithms provide extra complexity

6.3.3.5 Modification of Constants and Representations – M5

0 - not required

1 - required

6.3.3.6 New Functions – M6

0 – no new functions required

If new functions are required, the assessor must estimate how many will be involved.

n – there are n new functions to be created

6.3.3.7 Dual Currency Output – M7

Does this represent the internal processing ????????

0 – not required

If dual amounts have to be output, the assessor must estimate how many additional functions will be required to cater for this.

n – dual currency amounts must be output which means that n functions must be included/created

what does this function mean exactly and how does it relate to the print converters mentioned later in the list??

6.3.3.8 Interface Converters – M8

These are the interfaces to other applications and apply to the I/O Flows documented on DOC4 relating to Q4.8 and Q4.9 on DOC1.

0 – no interface converters are required
If interface converters are required, the assessor must estimate the related complexity level.

1 – interface converters are considered simple

2 – interface converters are considered complex

6.3.3.9 Archive Converters – M9

This is used to determine the effort required to convert the actual data in the archives. That is, the hardware aspect of the conversion. This will apply only to those countries whose NCUs do not currently contain decimals. This gives an estimate of the complexity of the archives to be converted.

0 – no converters required
If archives are to be converted, the assessor must estimate the complexity involved.
1 – archive converters will be simple *eg flat files???*
2 – archive converters will be complex *eg VSAM or DB??*

Suppose a customer has x simple converters and y complex ones, do we do two calculations and sum them or not????? Ie does this apply to the number of archives?

6.3.3.10 TP Input Converters – M10

How does this relate to the maps for TP programs???????

Ie is this the front end converter as described in solutions dup app, mask NCU, mask e??

0 – not required

If TP input converters are required, the assessor must estimate the complexity of them.

1 – simple converters required

2 – complex converters required

ie does this apply to the number of maps??

6.3.3.11 Print Converters – M11

How does this compare with M7

Ie is this the back end converter as described in solutions mask NCU, mask e??

0 – not required

If prints must be converted, the assessor must estimate how many are affected.

n – prints must be converted which means that n functions must be included/created

what does this function mean exactly??

6.3.3.12 Other Modifications – M12

0 – not required

n – other modifications result in n functions which must be included/created

6.4 Step 4 - Prepare the Estimate

Having obtained the information concerning the size and complexity of the system (and recorded it on DOC4) and the information concerning the modifications (and recorded it on DOC6) involved in the chosen ε solution, the assessor must now quantify the *Implementation Effort* (IE). This represents the effort required for the modifications to the application for the chosen solution and it must be recorded on DOC10, The Estimate (see section 5.11). Implementation Effort consists of *Basic Effort* (BE) plus *Extra Effort* (EE).

The *Basic Effort* for ε conversion depends upon the size of the conversion. That is, on number of programs (TP and Batch) and maps, within an application, which need modification.

The *Extra Effort* is related to the complexity of the conversion or peculiarities of the application in question. That is, on the type of ε solution and modification chosen. Extra effort for these other items must be calculated.

Finally, the estimate must be adjusted for contingency. This is done by adding a figure for *Other Effort (OE)*. This figure is calculated by applying the weightings calculated in step 6.2.2 to the Implementation Effort and the result recorded on DOC10, the Estimate.

To aid the calculations of the two figures to be recorded on DOC10, an Estimating Worksheet can be used. This is described in section 7.

This estimate for effort is used to cost the conversion project, underpin any contractual procedures and create the project plan. The project plan should contain exception procedures to be followed should unforseen circumstances arise.

6.5 Step 5 – Do In-use Evaluation

In use evaluation should be performed after every application has been converted to check that the effort parameters and weightings recorded in DOC2, DOC3, DOC7, DOC8 and DOC9 accurately reflect the capabilities of the personnel performing the conversion.

If there is any discrepancy, the reason must be determined and, if necessary, the parameters can be modified and the estimates for effort reworked as described in Step 1.

Where there is a large undesirable discrepancy, project exception plans must be followed.

7. Estimate Worksheet

While the various estimating forms can be 'programmed' via the use of Excel formulas, an Estimate Worksheet (see Table 2) has been supplied to demonstrate how to calculate the Implementation Effort (Basic Effort plus Extra Effort) and Other Effort required for ε conversion. The worksheet lays out the various efforts to be calculated and shows which documents contain the required figures/parameters.

7.1.1 The Basic Effort (BE)

The *basic effort* for ε conversion depends upon the number of programs (TP and Batch) and maps within an application which need modification. To calculate BE we multiply the number of each these by its *effort parameter* and sum the results. Experience has shown, however, that only a certain percentage will need modification. DOC7, Cost Item/Modification Parameters (see section 5.8), shows the percentage which will be likely to require modifications for each of the 11 modification types.

For example, if we are converting algorithms, at least 30% of TP programs and 50% of batch programs will require modifying.

Using the list of modifications documented in DOC6, the estimator now determines the percentage of programs and maps which will require modifications using the percentages documented in DOC7.

The basic effort is estimated using the adjusted number of programs and maps multiplied by a corresponding effort figure which takes into account the platform of the programs. The figures for effort are found in DOC8, Effort Parameters (see section 5.9), section 1 and the adjustment percentages can be found in DOC7, Cost Item/Modification Parameters (see section 5.8).

The basic effort covers Decimalisation (M1), Dual Currency (M2), Currency Code (M3), simple Algorithms (M4) and Constants (M5) and is calculated using the worksheet as:

$$\sum_{\text{All platforms}} (\#\text{Programs} * \% * \text{Effort Parameter})$$

Plus

$$\sum_{\text{All platforms}} (\#\text{Maps} * \% * \text{Effort Parameter})$$

The worksheet indicates the documents where the figures for each part of the calculation can be found.

7.1.2 The Extra Effort (EE)

The estimate for basic effort caters for size only. That is, it considers only programs and maps and not conditions of added complexity or peculiarities of the application in question. Extra effort will be required in addition to the Basic Effort to cater for these situations.

There are several conditions which cause added complexity and therefore extra effort:

- Archive modification (M9)
- Complex Algorithms (M4)
- Complex Converters (M8, M9, M10, M11).
- New Functions (M6, M7, M12)

Extra effort is calculated using the worksheet as:

The effort required for New Functions (FUNC)

Plus

The effort required for Extra Complexity (COMP)

Plus

The effort required for Converters (CONV)

Each of these is described below.

ESTIMATE WORKSHEET

Application	Example
Solution	Multi Currency
Modifications	M1, M3, M4, M5, M6, M9 & optionally M7, M8, M11

Basic Effort

	TP progs				Batch progs				Maps			
	#	%	eff	tot	#	%	eff	tot	#	%	eff	tot
Host	DOC4	DOC7	DOC8		DOC4	DOC7	DOC8		DOC4	DOC7	DOC8	
C/S	DOC4	DOC7	DOC8		DOC4	DOC7	DOC8		DOC4	DOC7	DOC8	
PC	DOC4	DOC7	DOC8		DOC4	DOC7	DOC8		DOC4	DOC7	DOC8	
Total												

Basic Effort (Programs + Maps)

Extra Effort

New Functions

	New Funcs			Dual Amt Funcs			Other Funcs		
	#	eff	tot	#	eff	tot	#	eff	tot
Host	DOC6	DOC8		DOC6	DOC8		DOC6	DOC8	
C/S	DOC6	DOC8		DOC6	DOC8		DOC6	DOC8	
PC	DOC6	DOC8		DOC6	DOC8		DOC6	DOC8	
Total									
Functions									FUNC

Extra complexity

	Algs			Archs			
	Code	eff	tot	#	%	eff	tot
1	DOC6	DOC8		DOC6	DOC7	DOC8	
2	DOC6	DOC8		DOC6	DOC7	DOC8	
Total							
Extra Complexity							COMP

Converters

	IO Flows (Interfaces)				TP Input Convs			Print Convs			Archs			
	#	%	eff	tot	#	eff	tot	#	eff	tot	#	%	eff	tot
1	DOC4	DOC7	DOC8		?	DOC8		DOC4	DOC8		?	?	?	
2	DOC4	DOC7	DOC8		?	DOC8		DOC4	DOC8		?	?	?	
Total														
Converters										CONV				

Extra Effort (New Functions + Extra Complexity + Converters)

Impementation Effort (Basic Effort + Extra Effort)

Other Effort

	Struct. Crit.	Maint. Comp.	Org. Imp	Doc. Avail.	Tech/Func. Ad	Other Costs %
L,M,H %						

Other Effort (Implementation Effort * Other Costs %)

Table 2 – Estimate Worksheet

7.1.2.1 Functions

New Functions

Are required if M6 (code>0) is selected on DOC6.

The code gives the number of new functions to be implemented.

The effort parameter for each is found in DOC8 and depends upon the platform.

New Function Effort is:

$$\sum_{\text{All platforms}} (\#\text{functions} * \text{Effort Parameter for platform})$$

Dual Amount Functions

Are required if M7 (code>0) is selected on DOC6.

The code gives the number of new functions to be implemented.

The effort parameter for each is found in DOC8 and depends upon the platform.

Dual Amount Function Effort is:

$$\sum_{\text{All platforms}} (\#\text{functions} * \text{Effort Parameter for platform})$$

Other Functions

Are required if M12 (code>0) is selected on DOC6.

The code gives the number of other functions to be implemented.

The effort parameter for each is found in DOC8 and depends upon the platform.

Other Function Effort is:

$$\sum_{\text{All platforms}} (\#\text{functions} * \text{Effort Parameter for platform})$$

7.1.2.2 Extra Complexity

Algorithms

Are to be modified if M4 (code>0) is selected on DOC6.

The code indicates whether algorithm modification is simple or complex.

The effort parameter is found in DOC8 and depends upon the defined complexity. It is a percentage of Basic Effort rather than a number of person days.

Algorithm Effort is:

$$(\text{BE} * \% \text{ for required complexity})$$

Archives

Archive formats are to be modified if M1 (code 2) is selected on DOC6.

The number of archive formats existing appears after the plus sign.

Not all archive formats for a given application will need to be modified for ε conversion. Experience has shown that only a certain percentage will need modification. The row labelled 'archives' on DOC7, Cost Item/Modification Parameters (see section 5.8), shows the percentage which will be likely to require modifications. This is a similar concept to the number of programs and maps required to estimate basic effort as described in section 5.8.

The effort parameter is found in DOC8 and depends upon the defined complexity.

Archive Format Conversion Effort is:

$$\left(\#\ *\ \%\ *\ \text{Effort Parameter for required complexity}\right)$$

Note – the percentage for simple archives is 0 indicating that this is catered for under Basic Effort.

7.1.2.3 Converters

Interface Converters

Are to be created if M8 (code>0) is selected on DOC6.

The code indicates whether interface modifications are simple or complex.

The effort parameter is found in DOC8 and depends upon the defined complexity.

Not all interfaces for a given application will need to be modified for ε conversion. Experience has shown that only a certain percentage will need modification. The row labelled 'interfaces' on DOC7, Cost Item/Modification Parameters (see section 5.8), shows the percentage which will be likely to require modifications. This is a similar concept to the number of programs and maps required to estimate basic effort as described in section 5.8.

The number of interfaces will be found on DOC4 as Interface Flows.

Interface Conversion Effort is:

$$\left(\#\ *\ \%\ *\ \text{Effort Parameter for required complexity}\right)$$

TP Input Converters

Are to be created if M10 (code>0) is selected on DOC6.

The code indicates whether the converters are simple or complex.

The effort parameter is found in DOC8 and depends upon the defined complexity.

The number of TP Input Converters will be found ??????????????

Where is this figure to be found?????????.

TP Input Converters Effort is:

$$\left(\#\ *\ \%\ *\ \text{Effort Parameter for required complexity}\right)$$

Print Converters

Are to be created if M11 (code>0) is selected on DOC6.

The code indicates whether the converters are simple or complex.

The effort parameter is found in DOC8 and depends upon the defined complexity.

The number of prints will be found on DOC4.

Print Converters Effort is:

$$\left(\#\ *\ \text{Effort Parameter for required complexity}\right)$$

Archive Converters

The actual data on the archives is to be converted using archive converters if M9 (code>0) is selected on DOC6.

The code indicates whether the converters are simple or complex.

The effort parameter is found in DOC8 and depends upon the defined complexity.

The number of archives will be found on DOC4.

Do we use the actual number or the adjusted number from archives above ???????????

Archive Converters Effort is:

$$(\#\ *\ \text{Effort Parameter for required complexity})$$

7.1.3 Other Effort (OE)

Is to be calculated using the weightings recorded on DOC4, IT Characteristics (see section 5.5). This form contains Low, Medium and High weightings for 5 cost items:

- Structural Criticality
- Maintenance Complexity
- Organisational Impact
- Documentation Availability
- Technical/Functional Adequacy

These weightings are used in conjunction with values recorded on DOC9, Other Costs Table (see section 5.10) to arrive at a percentage by which the estimate for Implementation Effort must be increased for contingency due to possible inadequacies in the baseline information.

$$\sum_{\text{All cost items}} (\text{Other Costs Increase \% for weighting for cost item})$$

The resulting percentage should be limited to a fixed upper percentage of 40% and lower percentage of 5%. This means, in effect, that other costs can not be less than 5% or more than 40% of implementation costs.

The implementation effort is multiplied by the resulting percentage to arrive at a figure for other costs effort.

Please triple check my archive calculations as I am sure that they can not be correct ☹

Appendices

The Renaissance Method, on which the εmergency method is based, is one which can be tailored to the re-engineering activities of any organisation. ε conversion, not being a true re-engineering exercise, is not fully compatible with Renaissance and therefore the activities in the Renaissance method do not map perfectly onto εmergency. However, by omitting some activities and redefining others, a satisfactory method for ε conversion can be achieved.

The Renaissance Method document is supported by 3 consultancy reports:

- Evolution Planning
- Modelling for System Evolution
- Client / Server Migration

Details of all four of the Renaissance reports can be found at

http://www.comp.lancs.ac.uk/computing/research/cseg/projects/renaissance/RenaissanceWeb/

Appendix E of this document contains a brief outline of the Renaissance method at the end of which there are three lists:

1. Lists the Renaissance activities and tasks described in the Renaissance Method report
2. Lists the Responsibilities and acronyms defined by the method with a cross reference to which activities they are related
3. Lists the tasks with a cross reference to which responsibilities are involved

In order to avoid unnecessary and costly assessment work, Renaissance recommends that system assessment be carried out iteratively at successively more detail omitting irrelevant detail at each iteration.

For εmergency two levels have been identified as relevant:

1. The Corporate Assessment
2. The Application Assessment

These assessments were described in section 2. Appendix B and Appendix C give details of how the Renaissance method was tailored to arrive at a method for εmergency at these two levels. Throughout these appendices, frequent references are made to the Renaissance activities, tasks and responsibilities. The Renaissance section numbers or acronyms, as listed in Appendix E, are used in these cases and are underlined.

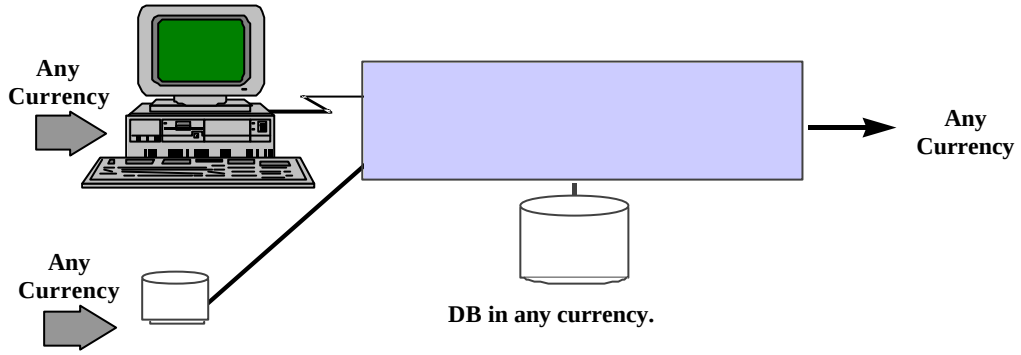
Appendix D contains templates for the documents devised for the εmergency method. Appendix A gives overviews of the ε solutions in more detail than in the body of this document along with tables to assist evaluation and selection of them.

8. Appendix A

This appendix deals with the possible solutions to EMU for an application of IT. It is intended to provide a set of decisional elements concerning the possible solutions without consideration for the peculiarities of the application to be adapted

8.1 MULTI-CURRENCY

A multi-currency application must be capable of handling any currency including ε, NCU and others such as the Yen and Dollar. It must, therefore, be capable of identifying and storing data in any currency. While storage can be in differing currencies, a base, or reference, currency must be used in order to produce a combined 'balance' figure.



Transforming a single-currency application to a ε compliant multi currency one requires large amendments to both the data and logic. These amendments can be summarised as follows:

- Introduction of parameters for the management of the multi-currency
  - Table reporting the characteristics of all the different currencies (code, description, decimal digits, rounding technique, etc)
  - Table with the exchange rate parameters
  - Table for conversion of labels
- Revision of user interfaces
  - Introduction of the currency code
  - Decimalisation of the amounts according to the handling of decimals and rounding for each currency
- Data base revision
  - Addition of the currency code to amount fields
  - Decimalisation
  - Conversion to the new format of current and historical data
- Revision of the application logic
  - Introduction to the logic of parameters enabling processing of all currencies
  - Interfaces with other TP and batch applications
  - Handling of peculiarities due to ε introduction (eg: in the transition period it is necessary to display the monetary amount both in the NCU and ε)

Considerations

Given the complexity and the effort required, the adoption of this solution must be motivated by functional requirements rather than by technical needs. That is, the introduction of the ε is seen as an opportunity to solve an existing problem.

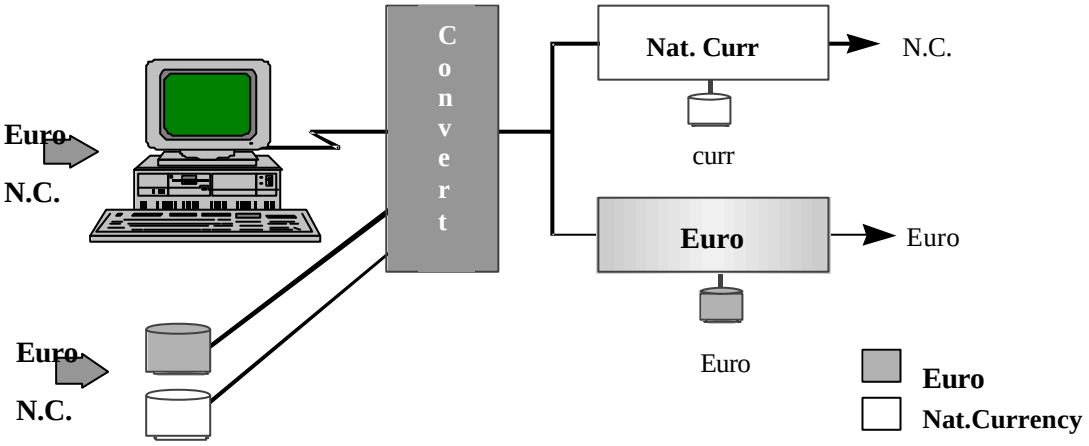
Advantages and Disadvantages

⇧	⇩
• Flexibility: Euro and future scenario • No rounding problem • Capability of handling the ε for countries not participating	• Huge manual modification • High costs • Not required for application operating in one currency only (majority)

8.2 DUPLICACTE APPLICATION

This solution, once implemented, should be able to manage both the transition and the following period. It involves:

- Duplication of the application (clone)
- Introduction and management of the second currency (ε)
- Change of interface and archives only to manage the two currencies



Activities required to implement this solution are:

- Implementation of the conversion functions
- Adaptation of the application in NCU to ε
- Revision of the accounting system

Considerations

This solution is particularly suitable for those applications for which it is possible to foresee the use of the NCU only up to a given date. It is also convenient for those applications that do not require the execution of cross currency operations (e.g: operations in NCU on accounts in ε). It allows a progressive change from NCU to the ε.

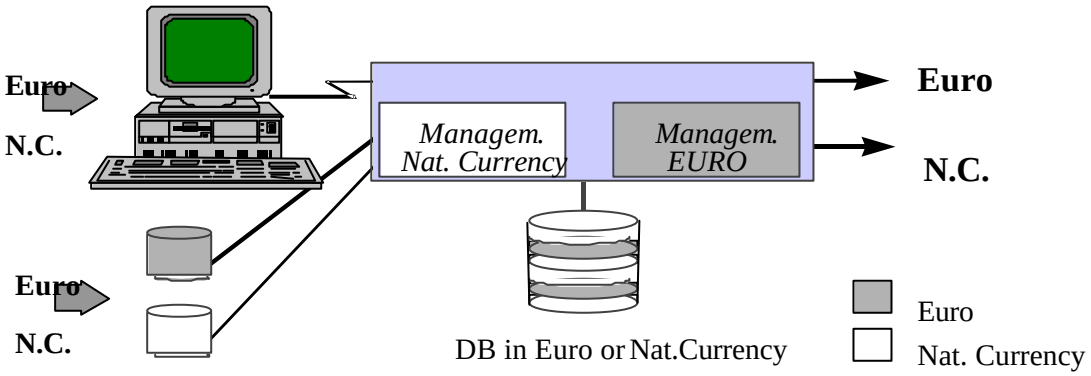
Advantages and disadvantages

⇧	⇩

• Changes are not confined to the transition period
• They may be implemented incrementally
• The impact is limited to the Euro version only
• The new application does not require many functions for managing the transition period
• The rounding problem is limited to the interface functions
• The adaptation can be highly automated
• If required, the conversion of historical archives can be postponed until after the transition period
	• Need to manage two application:
	  • manintenance
	  • scheduling
	  • interfaces
	• The "conversion" process, if not completely automated, may be heavy from the organisation point of view.
	• Need to manage cross currency operations

8.3 DUAL CURRENCY

This solution involves the ability of the application to input, output and store monetary amounts in either ε or NCU with a code to distinguish which. However, wherever possible, outputs will remain in NCUs. *Is this right???*



This is a simplified version of the multi currency solution and, providing it is in place within 1999, allows progressive introduction of the ε.

The adoption of this solution requires:

- Revision of maps and reports (U.I.)
  - Introduction of the "currency code" for all input and output
  - "decimalisation"
- DB Revision
  - Introduction of fields for the currency code for all data structures representing monetary amounts

    Dario had 'dates' here originally – assume it was a misprint!!!!
  - "decimalisation"
  - Conversion of archives to the new format

- Revision of the application logic
  - Management of the peculiarities of the two currencies
  - Management of constants describing the currency
  - Interface with other TP and batch applications.
  - Adaptation to manage peculiarities of ε (eg. Rounding policy, visualisation of two currencies in the transition period).
  - Introduction of a conversion function (ε <-> NC) where necessary (syncronisation points)

Considerations

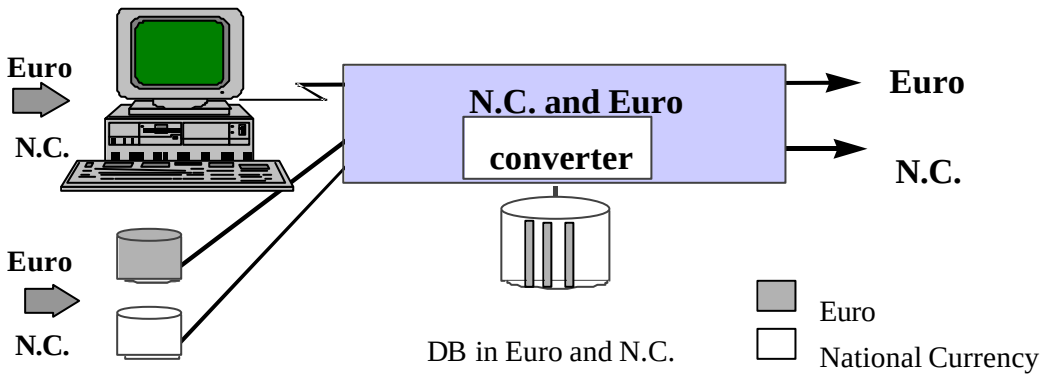
This solution facilitates the management of the transition period and enables customers to familiarise themselves with ε amounts in a progressive manner.

Advantages and disadvantages

⇧	⇩
• Progressive introduction of ε • Except for specific cases, there is no rounding problem	• Need to operate in two phases • Impact on the operating applications • Need to complete the work within 1999 • Need to manage cross operations • The changes are mainly manual

8.4 DUAL AMOUNTS

This solution involves the ability of the application to input and output monetary amounts in either ε or NCU, as for Dual Currency above, but storing both the amounts. This means that input is always duplicated and will need handling consistently.



The application will output amounts in the two currencies either by two amounts on one output or by a duplicate output.

The adoption of this solution requires:

- Revision of maps and reports (U.I.)
  - Introduction of the ε amount somewhere
  - “decimalisation”

- DB Revision
  - Introduction of additional amount field for all data structures representing monetary amounts
  - "decimalisation"
  - Conversion of archives to the new format
- Revision of the application logic
  - Management of the peculiarities of the two currencies
  - Management of constants describing the currency
  - Interface with other TP and batch applications.
  - Adaptation to manage peculiarities of ε (eg. Rounding policy, visulaization of two currencies in the transition period).
  - Introduction of a conversion function (ε <-> NC) where necessary (syncronisation points)

Considerations

This solution facilitate the management of the transition period and familiarises customers to ε amounts in a progressive manner

From the implementation point of view, the solution is best applied to those applications with a limited number of monetary fields in maps and reports and on the data base.

Advantages and Disadvantages

⇧	⇩
• Progressive introduction of ε • No rounding problems: the amounts are expressed in both the currencies	• Need to operate in two phases • Impact on the operating application • Need to complete the work within 1999 • The changes are mainly manual • More disk space required

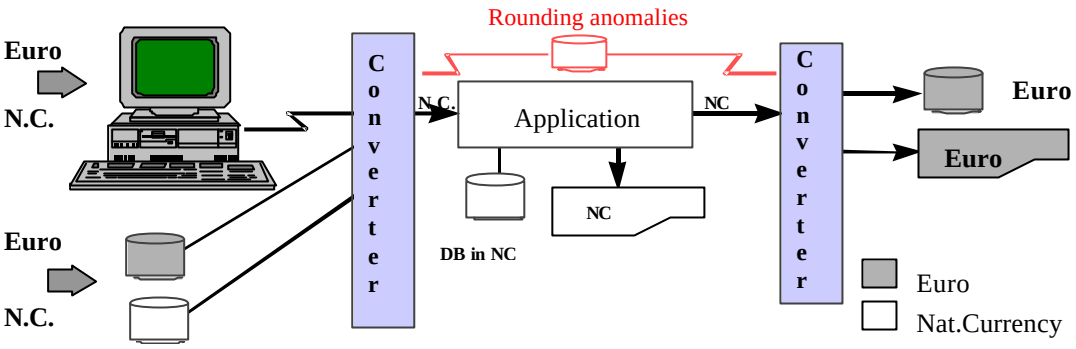
8.5 MASKING NATIONAL CURRENCY

This solution requires modifying an application to allow for direct input of NCUs but input of εs through the use of a converter. Storage is in a reference currency and output for ε, if required, is via a converter. This is a temporary solution for the transition period and is applicable only when the number of monetary amounts being handled is small.

Darios write up appears to contradict the diagram!!!

It means that changes to the database are kept to a mimimum and that output reports and flows can be converted to as required.

The adoption of this solution may generate different scenarios depending on the chosen reference currency.



The solution is characterised as follows:

- The original application, operating in NCU, remains the same
- All the amounts input must be in NCU. Where input is in ε the converter is applied
- All the amounts output are in NCU. Where interfaces with other applications are needed during the transition period a converter may be used.

The adoption of this solution requires:

- Revision of maps and reports (UI)
- Revision of the application logic

Considerations

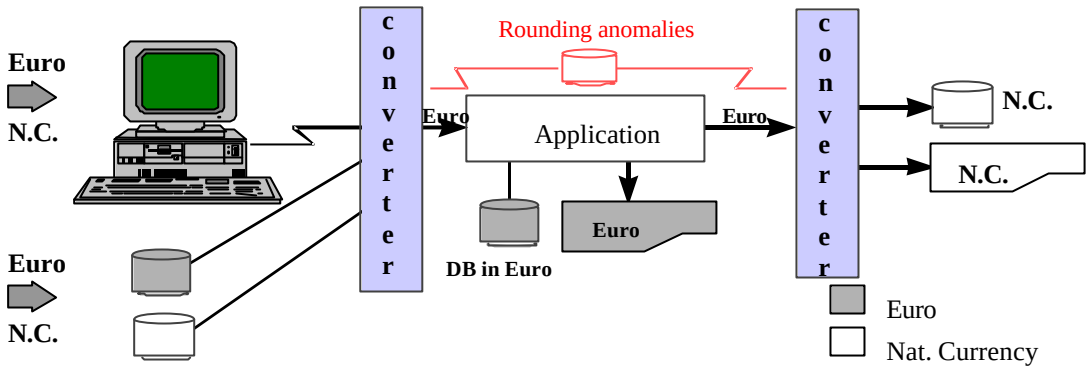
This is a temporary solution for the transition period. It is applicable only when the number of input amounts is small. Additionally some internal computations are necessary.

Advantages and Disadvantages

⇧	⇩
• Simple analysis and implementation. The effort will be mainly in the converter • Limited changes to the applications with the possibility of using tools • Delay the final solution	• Temporary solution only • Rounding problems must be handled • A substantial amount of further work will be required at the end of the transition period

8.6 MASKING IN EURO

This is similar to the previous solution except the amount stored is in ε.



The solution is characterised as follows:

- The original application, operating in NCU, is converted into "single-currency" ε.
- All the monetary amounts input must be in ε. However, where input is in NCU, a converter to transform it can be used.
- All monetary amounts output are in ε. To interface with other applications and the peculiarities of the transition period a converter may be used.

The adoption of this solution requires:

- Conversion of the original application to ε
- Use of converters

Considerations

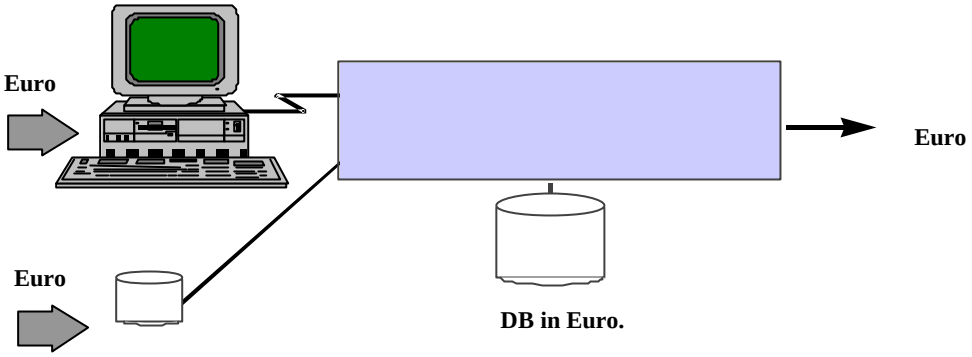
While this solution solves post transition period problems it is based on the adaptation of the original application from NCU to ε. This means that, though it is quite a complex and expensive solution, it is permanent even after the transition period when it is a simple process to remove the converters.

Advantages and disadvantages

⇧	⇩
• Results in an application in ε also suitable for the transition period • Removal of converters is simple	• Its completion is required before the commencement of the transition period

8.7 SINGLE CURRENCY: EURO

This solution requires the transformation of an application operating in NCU only to operate in ε only. The changes impact all application components (user interfaces, DB, programs).



In particular, the following changes are required:

- Revision of maps and reports (UI)
  - "decimalisation" of the amounts
- Revision of DB
  - Amount fields formats
  - Conversion of existing DB as for format and data (migration)
- Revision of the application logic to manage ε
  - "decimalisation" of amounts in programs
  - revision of the constants related with amounts
  - revision of computational parts of programs to reflect the differences between the two currencies (eg: rounding policy)

Considerations

This is the most appropriate solution to the conversion problem. However, Its adoption, requires further action to manage the transition period. The adaptation can be supported by tools.

Advantages and Disadvantages

⇧	⇩
• Most appropriate solution • Can be supported by tools	♦ Requires additional action during the transition period

8.8 Selection of the solution

Looking at the 7 solutions outlined in the previous section we can say that none is completely suitable for all applications supporting an organisation's IT base. The choice of a solution, therefore, must be made by considering the characteristics of a specific application. The following should be considered:

- *Application Duplication* should be adopted where ever possible as it:

- Minimises costs
- Is not limited to the transition period
- Does not require a team with in depth knowledge of the application
- Applications that may be adapted using this solution are:
  - Ones that can continue to operate in the national currency during the transition period (no external constraints)
  - Ones that can operate on duplicated DB

- *Masking in Euro* allows an application to be adapted and, at the same time, to cover the transition period.
- *Multi-currency* should be adopted only for those applications already operating in a multi-currency logic or if external needs (market opportunities, norms) so dictate. The adoption of this solution is unlikely to be widespread as it will probably require the complete re-design of an application.

8.9 Evaluation of the solution

Table 4 summarises the main characteristics of the 7 solutions and uses the following evaluation parameters:

Parameter	Description
Implementation	Cost of the implementation
Dead code	Dead code required to manage the transition period and to be removed later.
Rounding	Management of rounding residues due to ε conversion and the maintenance of balances in a reference currency
Tools	Possibility of using tools to adapt application source
Converters	Need for use of converters
Transition	problem related only with the transition period
Final	Solution usable after the transition period

Table 3 – Solution Evaluation Parameters

SOLUTION	PARAMETERS							NOTES
	Implemen-tation	Dead code	Rounding	Tools	Converter	Transition	Final	
multi-currency	Very High	✗	✗	✗	✗ (*)	✓	✓	To be applied only if strictly required
Duplicate Application	Medium	Low	✗	✓	✗ (*)	✓	✓	Optimal
Dual Currency	High	High	✓	✗	✓	✓	✓	Generally valuable
Dual Amounts	Medium	Medium	✗	✓	✓	✓	✓	With few amounts only
Masking National Currency	Low	Low	✓	✗	✓	✓	✗	Temporary: to be used if the application will be replaced after the transition period
Masking Euro	High	Low	✓	✓	✓	✓	✓	Optimal
Single-currency: Euro	Medium	Low	✗	✓	✗ (*)	✓	✓	Require some work to manage the transition period (converters)

(*) Converter may be required for DB migration and to manage eventual additional functionalities.

Table 4 – Solution Characteristics

9. Appendix B
Tailoring Renaissance - Corporate Level

This appendix describes the Renaissance method tailoring for the Corporate Assessment. At this level, the purpose of the method is to devise a rough order of magnitude for ε conversion effort and provide enough information to enable a master plan for the Application Assessment to be created (see section 6.1).

9.1 Method Startup (2.3.1)

The Corporate Assessment is specified and the method commences with an empty repository. In this instance, the εmergency team performed part of the method startup. Method startup creates the baseline from which all subsequent iterations of the method will start.

9.1.1 Create Repository Baseline (2.3.1.1)

This was performed by the εmergency team and involved selecting the tasks to be performed from the list of tasks in the method outline and stipulating responsibilities and the artifacts to be produced. There are two types of tasks to be performed at this level:

- Those to be performed prior to the assessment in order to plan the assessment 2.3.3.1, 2.3.3.2
- Those to be done as part of the assessment 2.3.2, 2.3.3.3, 2.3.3.4, 2.3.4, 2.3.5.3.1

For εmergency 2 responsibilities have been created to perform the tasks:

- Assessor
- Customer

There may be many people performing assessor and customer responsibilities. Amongst them they provide the expertise of ABE, CL, EM, EST.

9.1.2 Identify Constraints and Business Goals (2.3.1.2)

This task will be performed by the assessor who will liaise with the customer to document an εvolution strategy and corporate constraints.

For example, “We want an estimate for a mono->multi currency conversion in order to satisfy business goals x, y and z. We will convert our General Ledger systems at time t1 and the others at time t2 forgetting about dual displays. We want it tested like this; we have spare testing capacity only in the evenings; we have limited staff availability”.

9.2 Business Processes (2.3.2)

The assessor who will model, in some way, the different business functions of the organisation to identify applications to be assessed performs this task.

9.3 Assess Current Situation (2.3.3)

This assessment is a higher level than envisaged in Renaissance. At this level we are merely attempting to identify and list the applications serving the business and establish a rough idea of their size to enable the ball park estimate of effort to be created and enable the master plan for the Application Assessment to be completed. We are not assessing the IT systems of an organisation for fitness for reengineering therefore we have no need to look at Technical Quality and Business Value. We have no option but to transform any system handling monetary values to make it ε compliant. It may be high Business Value systems should be assessed first but it is probably not necessary to do a full business value analysis to determine this fact.

9.3.1 Specify Assessment Levels (2.3.3.1)

The corporate level for this iteration was specified at startup time. At this level, we must look at applications as black box entities in order to create the foundations of a catalogue of applications.

9.3.2 Select Assessment Characteristics (2.3.3.2)

This task was performed by the emergency project team and resulted in a document to be used to catalogue the applications supporting an organisation's IT base (see section 5.1). This catalogue forms the base for the repository in emergency and is completed as part of step 2.3.3.3.

The following characteristics are relevant at this level:

Characteristic	Assessment
Application Identifier	N/A
Business Area	Financial (General Ledger / Stock Control / Payroll etc), Non-financial. Given this it is possible to estimate the number of money fields according to a set of parameters for use in application assessment.
Type	In house / package etc. Given this it is possible to identify who is responsible for the conversion.
Personnel	A list of the customer personnel with the required knowledge of the application to be contacted during the Application Assessment. If no one is available to supply such information, an organisation must consider redevelopment or replacement
Hardware platform	This is a list of platforms which the application uses
Size and complexity	A quick measure for application size and complexity. The larger and more complex the system is, the more time will be needed to gather estimating information and convert and test it regardless of how many money fields it contains. Guesstimate (or ball park figure) of small, medium and large

Table 5 - Corporate Assessment Characteristics

9.3.3 Assess Characteristic (2.3.3.3)

The assessor performs this with assistance from a customer representative. This task will generate

- staffing requirements –
  - assessor
  - customer representative
    At this level, the personnel allocated must be familiar with the IT base of the organisation performing the assessment but not necessarily have any in depth understanding of any particular application
- artifacts – the artifact generated is that listed in section 5.1 – DOC0

This assessment involves identifying applications and recording them in the catalogue with the necessary characteristics.

9.3.4 Overall Assessment (2.3.3.4)

The assessor who will apply the parameters to the applications to arrive at a rough order of magnitude for the conversion effort for each will perform this. The overall assessment for ε conversion will be the sum of the order of magnitude assessments for all applications identified.

9.4 Model Context of Current System (2.3.4)

This will be performed by the assessor with assistance from the customer if necessary. This is really a combination of this task and 2.4.3.1 and is performed in parallel with 2.3.3. It assists in mapping applications to business functions and enables situations such as that described in section 3.1 to be identified.

9.5 Select Target (2.3.5)

This activity, with all its sub tasks, is intrinsically different from Renaissance. For ε conversion it is concerned with the strategical prioritisation of the identified applications. It facilitates a way of quantifying the assessment effort and planning the subsequent Application Assessment.

9.6 Plan Evolution Project (2.4.1)

This task will use the identified applications to create a master plan for the Application Assessment bearing in mind the prioritisation in step 2.3.5. In addition, the assessor will devise the effort parameters and weightings to be used in the Application Assessment.

artifacts – the artifacts generated are those listed in sections 5.3, 5.4, 5.8, 5.9, 5.10 – DOC2, DOC3, DOC7, DOC8, DOC9

9.7 Tasks 2.4.2 to 2.6

Are not required for the Corporate Assessment.

10. Appendix C
Tailoring Renaissance - Application Level

This appendix describes the Renaissance method tailoring for the Application Assessment. At this level, the purpose of the method is to devise a reasonably accurate estimate for ε conversion effort for an application so that a conversion project plan can be formulated and possible contractual procedures established. At this level the tasks (see sections 6.2 to 6.5) are being performed in accordance with the master plan devised in the Corporate Assessment. The actual effort to run the method to assess any application can be monitored and used to re estimate effort for subsequent Application Assessments (see section 9.6).

10.1 Method Startup (2.3.1)

The Application Assessment is specified and begins with the catalogue of applications created during the Corporate Assessment. Each of these, which require conversion for ε, is assessed at this level.

10.1.1 Create Repository Baseline (2.3.1.1)

This was performed by the εmergency team and involved selecting the tasks to be performed from the list of tasks in the method outline and stipulating the artifacts to be produced. As with the Corporate Assessment, there are two types of tasks to be performed at this level:

- Those to be performed prior to the assessment in order to plan the assessment 2.3.3.1, 2.3.3.2

- Those to be done as part of the assessment
  2.3.2, 2.3.3.3, 2.3.3.4, 2.3.4, 2.3.5

For εmergency 2 responsibilities have been created to perform these tasks:

- Assessor
- Customer

There may be many people performing assessor and customer responsibilities. Amongst them they provide the expertise of ABE, CL, EM, EST, LFE, LIE, LSOM, SWE, SWPM, U.

10.1.2 Identify Constraints and Business Goals (2.3.1.2)

The assessor who will liaise with the customer will perform this task. Constraints and Business Goals will be present from the Corporate Assessment but may need to be enhanced in the light of the application characteristics identified. For example it may be necessary to plan for contracting 'agency' personnel for the conversion tasks where organisation personnel fall short of identified requirements.

10.2 Business Processes (2.3.2)

The assessor to enhance, if necessary, the information recorded at the Corporate Assessment performs this task.

10.3 Assess Current Situation (2.3.3)

At this level we are attempting to establish application characteristics which enable an estimate ε conversion effort to be created. We are not assessing IT systems for fitness for reengineering therefore we have no need to look at Technical Quality and Business Value. We have no option but to transform any system handling monetary values to make it ε compliant. We need to assess the work content involved in the conversion based on an application in the catalogue and the εvolution strategy defined as a constraint in the method startup. It may be that high Business Value systems should be assessed first but it is probably not necessary to do a full business value analysis to determine this fact.

10.3.1 Specify Assessment Levels (2.3.3.1)

The Application Level for this iteration was specified at startup time. At this level we will look at the cost items (see section 3.7) for each application.

10.3.2 Select Assessment Characteristics (2.3.3.2)

This task was performed by the εmergency project team and resulted in a set of documents to be used to record application, solution and effort information for any application (see section 5.2 to section 5.11). These serve to flesh out the catalogue previously devised and are completed as part of step 2.3.3.3.

The estimate is being based on application size and complexity and adjusted for uncertainty. Application characteristics relevant at this level, and the aspects of each to be quantified, are given in Table 6.

Characteristic	Assessment Aspect
Documentation	Availability Type State
Technical / Functional Adequacy	Revisions – when and why Business Processes supported Quality as perceived by the end user

	Technical quality
Organisational Impact	Number of departments using the application Number of users using the application Deployment method
Maintenance Complexity	Age Revisions when and why Hardware and software platform Maintenance effort expended in the last year Maintenance effort forecast for the coming year
Structural Criticality	Number of programs (batch and on line) Number of maps Number of archives Number of prints Number of I/O flows
Information Reliability	Level of responsibility of customer contact Technical competence of customer contact Seniority of customer contact Availabililty of users Personal impression of assessor

Table 6 - Application Assessment Characteristics

10.3.3 Assess Characteristic (2.3.3.3)

The assessor performs this with assistance from a customer representative for the application being assessed.

This task will generate

- staffing requirements –
  - assessor
  - customer representatives
    at this level, the personnel allocated must be familiar with the hardware, support software and application programs utilised by the particular application chosen. Their knowledge must be deeper but not necessarily as broad as the personnel allocated to the Corporate Assessment.
- artifacts – the artifacts generated are those listed in sections 5.2, 5.5 and 5.7

This assessment involves:

- Documenting the information concerning the characteristics defined – DOC1
- Establishing application criticality and completing the required form – DOC4
- Establishing the modifications required to satisfy the ε solution and completing the required form – DOC6

10.3.4 Overall Assessment (2.3.3.4)

The assessor who will apply the effort parameters and weightings to the cost items to generate an estimate of effort performs this. The parameters and weightings applied are those which were devised as part of step 2.4.1 (see section 9.6).

- artifacts – the artifact generated is that listed in section 5.11 – DOC10

10.4 Model Context of Current System (2.3.4)

The assessor will perform this with assistance from the customer if necessary. This is really a combination of this task and 2.4.3.1 and is performed in parallel with 2.3.3. It assists in identifying the dependencies between applications and enables interfaces to other applications to be identified.

10.5 Select Target (2.3.5)

This task is intrinsically different from Renaissance. For ε conversion this step will be concerned with refining the strategical prioritisation of the identified applications. It facilitates a way of planning the actual conversion projects and providing a base for establishing formal contracts.

10.6 Plan Evolution Project (2.4.1)

This task is purely as envisaged in Renaissance. That is planning the work of the actual conversion project.

10.7 Perform re-engineering (2.4.2 – 2.6.4)

These tasks are purely as envisaged in Renaissance.

10.8 Do In-Use Evaluation (2.6.5)

As each conversion project is completed, the figures for actual effort for conversion can be fed back into the Corporate Assessment in order for the effort parameters to be amended (see section 9.6) and the conversion re-estimated. This in turn may have an impact on project planning.

11. Appendix D
Documents for Repository

Id	Name	Appendix
DOC0	Application Catalogue	D1
DOC1	Customer Questionnaire	D2
DOC2	Assessor Questionnaire	D3
DOC3	Direct Effort Weights	D4
DOC4	IT Characteristics	D5
DOC5	Modification/Solution Table	D6
DOC6	Euro Solution Characteristics	D7
DOC7	Cost Item / Modification Parameters	D8
DOC8	Effort Parameters	D9
DOC9	Other Cost Parameters	D10
DOC10	Estimate	D11

11.1 DOC0

We need a simple spreadsheet for this – do you want me to do it or do you want to do it in the light of your experiences??

11.2 Appendix D2 – DOC1 – Customer Questionnaire

EURO ASSESSMENT
Information Collection Questionnaire for Software Applications

CUSTOMER QUESTIONNAIRE

COMPANY: _____

DEPARTMENT or SYSTEM: _____

SUB-SYSTEM: _____

APPLICATION: _____

APPLICATION ACRONYM _____

Date of the interview :_____

Name of the interviewer :_____

1° interviewed :_____

Role :_____

Phone and fax :_____

2° interviewed :_____

Role :_____

Phone and fax :_____

3° interviewed :_____

Role :_____

Phone and fax :_____

SEC. 1. GENERAL DATA

1.1. Brief **description** of the application**.**	_____ _____ _____
1.2. **Departments/Personnel** responsible for the application	Organisation _____ User _____ EDP _____
1.3. **Type** of application?	1. ☐ Developed in-house; development started on _____ 2. ☐ Package acquired after a study started on _____ 3. ☐ Developed by a third party; development started on _____ 4. ☐ Outsourcing
1.4. When was application **deployed**?	Date (mm/yyyy) _____
1.5. **Last revision** date	Date (mm/yyyy) _____
1.6. Is **replacement/redevelopment** of the application foreseen shortly? IF YES, • When? • For which reasons • With which approach?	1. ☐ Yes 2. ☐ No _____ 1. ☐ Technical reasons 2. ☐ functional reasons 3. ☐ Other _____ 1. ☐ in-house development 2. ☐ Acquisition of a package 3. ☐ Development contracted to a third party 4. ☐ The application will be given in outsourcing

1.7. Is the application under **revision**? IF YES, • When? • For which reason?	1. ☐ Yes 2. ☐ No _____ 1. ☐ Technical reasons 2. ☐ Functional reasons 3. ☐ Others _____
1.8. Is **documentation** available?	1. ☐ Yes 2. ☐ No If Yes, which? / Is it up to date? 1. ☐ Functional Analysis — 1. ☐ Y 2. ☐ N 2. ☐ Technical Analysis — 1. ☐ Y 2. ☐ N 3. ☐ Data Dictionary — 1. ☐ Y 2. ☐ N 4. ☐ User Manual — 1. ☐ Y 2. ☐ N 5. ☐ Administration Manual — 1. ☐ Y 2. ☐ N 6. ☐ Other:_____ — 1. ☐ Y 2. ☐ N
1.9. **EURO** compliancy assessment: ♦ is a currency code included in archives? ♦ Do monetary amounts in archives contain decimal digits? ♦ Is the application multi-currency? ♦ In your opinion, is the application EURO compliant already? IF NO, how broad do you consider the impact due to the EURO adaptation?	 1. ☐ Yes 2. ☐ No 1. ☐ Yes 2. ☐ No 1. ☐ Yes 2. ☐ No 1. ☐ Yes 2. ☐ No 1. ☐ High 2. ☐ Medium 3. ☐ Low WHY? _____

1.10. **Functional Coverage**: number of enterprise business processes supported by the application	1. ☐ High 2. ☐ Medium 3. ☐ Low
1.11. **Functional Quality**: application quality as perceived by the end user of specific enterprise functions	1. ☐ High 2. ☐ Medium 3. ☐ Low
1.12. **Technical Quality:** adequacy of the adopted technology and compliance level with respect to company standards	1. ☐ High 2. ☐ Medium 3. ☐ Low

SEC. 2. INFORMATION RELATED WITH PACKAGES

2.1. **Name** of the package	_____
2.2. Name of the **provider**	_____
2.3. **Purchase** contract type	_____
2.4. **Level of customisation** and/or integration:	1. ☐ none 2. ☐ marginal 3. ☐ consistent____% (<20% tot cost) (>20% tot cost)
2.5. Customisation	1. ☐ in house 2.☐ provider 3. ☐ third party
2.6. Package **version and release** date	_____
2.7. **Last version** available on the market	
2.8. Type of **maintenance** contract	_____ 1. ☐ none 2. ☐ ord.ry main. 3. ☐ Extraord.ry main.
2.9. **Is EURO** adaptation under the maintenance contract? If Yes, what is the cost?	1. ☐Yes 2. ☐ No _____
2.10. Are **sources** available?	1. ☐ Yes 2. ☐ No

SEC. 3. TECHNICAL ASPECTS

3.1 Is the application **Client/Server?**	1. ☐ Yes 2. ☐ No If Yes, with how many levels? _____
3.2 Which is the running **environment**?	**HOST** _____ HW _____ S.O _____ TP Monitor_____ DBMS _____ Prog. Lang.1 _____ _____% Prog. Lang.2 _____ _____% Tools _____ **DEPARTIMENTAL** _____ HW _____ S.O _____ TP Monitor_____ DBMS _____ Prog. Lang.1 _____ _____% Prog. Lang.2 _____ _____% Tools _____ **PC** __ _____ HW _____ S.O _____ TP Monitor_____ DBMS _____ Prog. Lang.1 _____ _____% Prog. Lang.2 _____ _____% Tools _____

3.3. Does usage of the application involve other **equipment** (HW) affected by the currency change? (ATM, POS, cash-counter, ...)	1. ☐ Yes 2. ☐ No Number Type _____ _____
3.4. **Number** (or estimate) of source **programs**	TP Pgms Maps BATCH JCL _____ _____ _____ _____ 1. ☐ estimate 2. ☐ exact
3.5. **Number** (or estimate) of **statements**	TP BATCH _____ _____ 1. ☐ estimate 2. ☐ exact
3.6. **dead-code** information: • % of unused **statements** • % of unused **programs**	 _____% _____%
3.7. **Number** (or estimate) of **archives**.	Traditional Archives _____ DB2 Tables _____ DL/I Segments _____ Other DBMS _____ 1. ☐ estimate 2. ☐ exact
3.8. **per./months** effort last year for **maintenance** of the application 3.9. **per./months** effort forecast for **maintenance** during the current year For Packages fully maintained by the provider set the maintenance effort to ZERO	_____ _____

SEC. 4. FUNCTIONAL ASPECTS

4.1. How many **Departments** use the application?	_____
4.2. How many end **users** of the application are there?	_____
4.3. How is the application used?	1. ☐ by internal staff for internal purposes 2. ☐ by internal staff to support the provision of services to clients 3. ☐ by clients directly
4.4. What **time-window** is used by the application?	1. ☐ less than one month 2. ☐ between one month and one year 3. ☐ between one and three years 4. ☐ more than three years
4.5. What time-window is relevant to the application for **archiving and management of historical data**?	1. ☐ less than one year 2. ☐ between one and five years 3. ☐ more than five years
4.6. What deployment **method** was used for the application?	1. ☐ deployment for all users at the same time 2. ☐ deployment to groups of users number of months for the deployment due to the second scenario _____
4.7. Nr. **Prospects/Modules/Prints** • internal users • external users (eg. clients)	Blank paper/flash Forms _____ _____ _____ _____
4.8. Number of **I/O FLOWS** to other applications imposed by the company: • from/to internal applications • from/to applications of companies in the same Group • from/to external applications	INPUT OUTPUT _____ _____ _____ _____ _____ _____

4.9. Number of **I/O FLOWS** to other applications imposed by other organisations (eg BACS): • from/to applications of other companies in the Group • from/to external applications	INPUT OUTPUT _____ _____ _____ _____
4.10. How many other **applications** may be **impacted** by modifications to the one under consideration?	Owned by the Company _____ _____ _____ _____ Owned by other Company in the Group _____ _____ _____ _____
4.11. Do you know any application, developed internally, that may be impacted by modifications to the one under consideration? If Yes, • managed internally? • managed by other companies of the Group?	1. ☐ Yes 2. ☐ No 1. ☐ Yes 2. ☐ No 1. ☐ Yes 2. ☐ No If possible, list them: Office Application Environment _____ _____ _____ _____ _____ _____

11.3 Appendix D3 – DOC2 – Assessor Questionnaire

USER RELIABILITY: To be filled in by the interviewer and used to adjust weights

Level of responsibility on the application	
direct	0
supervision	7
none	14
Technical competence	
high	0
medium	12
low	24
seniority in the role	
more than 5 years	0
between 2 and 5 years	12
less than 2 years	24
availability of the user	
complete	0
medium	7
scarce	14
personal impression by the interviewer	
positive	0
sufficient	12
negative	24

USER RELIABILITY		adjustment factor for weights
LOW (L) between 0 and 33	33	30
MEDIUM (M) between 34 and 66	66	15
HIGH (H) more than 66	100	0

11.4 Appendix D4 – DOC3 – Direct Effort Weights

Parameter Description			WEIGHT	
Documentation Availability (question 1.8)				
Functional Analysis				
	Not available		0	
	Not updated		7	
	Updated		15	
Technical Analysis				
	Not available		0	
	Not updated		12	
	Updated		25	
Data Dictionary				
	Not available		0	
	Not updated		7	
	Updated		15	
User Manuals				
	Not available		0	
	Not updated		7	
	Updated		15	
Administration Manuals				
	Not available		0	
	Not updated		10	
	Updated		20	
Other Manuals				
	Not available		0	
	Not updated		5	
	Updated		10	
Total Percentage	LOW (L)	**>zero<=**	33	plus (result "USER RELIABILITY")
	MEDIUM (M)	**>L, <=**	66	plus (result "USER RELIABILITY")
	HIGH (H)	**>M, <=**	100	
Technical/Functional Adequacy				
Application Revision (question 1.7)				
	foreseen for technical reasons		5	
	foreseen for functional reasons		5	
	foreseen for other reasons		15	
	not foreseen		20	

Functional Coverage (question 1.10)

high	20
medium	12
low	0

Functional Quality (question 1.11)

high	20
medium	12
low	0

Technical Quality (question 1.12)

high	20
medium	12
low	0

Total Percentage	LOW (L)	**>zero<=**	33	plus (result "USER RELIABILITY")
	MEDIUM (M)	**>L, <=**	66	plus (result "USER RELIABILITY")
	HIGH (H)	**>M, <=**	100	

Organisational Impact

Units using the application (question 4.1)

from 1 to 10	10
from 11 to 30	30
more than 30	40

Users using the application (question (4.2)

from 1 to 20	10
from 21 to 60	30
more than 60	40

Deployment Policy (question 4.6)

in one shot to all groups and users	0
for groups of users in less than 3 months	10
for groups of users in more than 3 months	20

Total Percentage	LOW (L)	**>zero<=**	33	minus (result "USER RELIABILITY")/3
	MEDIUM (M)	**>L, <=**	66	minus (result "USER RELIABILITY")/3
	HIGH (H)	**>M, <=**	100	

Maintenance Complexity

Initial Deployment date (question 1.4)

more than 5 years	15
between 1 and 5 years	6
less than 1 year	0

Application Replacement Shortly (question 1.6)

foreseen for technical reasons	10
foreseen for functional reasons	10
foreseen for other reasons	20
not foreseen	0
Operational Environment (question 3.1; 3.2)	
host application	10
C/S application	5
PC application	3
maintenance effort in the last year (question 3.8)	
more than 5 months	35
between 3 and 5 months	20
between 1 and 3 months	10
less than 1 month	0
maintenance effort foreseen this year (question 3.9)	
more than 5 months	20
between 3 and 5 months	10
between 1 and 3 months	5
less than 1 month	0

Total Percentage	LOW (L)	**>zero<=**	33	minus (result "USER RELIABILITY")/3
	MEDIUM (M)	**>L, <=**	66	minus (result "USER RELIABILITY")/3
	HIGH (H)	**>M, <=**	100	

Structural Criticity

Number of Programs (question 3.4)	
less than 50	5
from 50 to 99	10
more than 99	20
Number of Maps (question 3.4)	
less than 10	1
from 20 to 50	5
more than 50	10
Number of Archives (question 3.7)	
less than 10	3
from 20 to 50	8
more than 50	15
Number of Prints (question 4.7)	
less than 10	1
from 20 to 50	3
more than 50	5
Number of I/O FLOWS (questions 4.8; 4.9)	
less than 10	3

	from 20 to 50			8	
	more than 50			15	
Total Percentage	LOW (L)	**>0,**	**<=**	33	minus (result "USER RELIABILITY")/3
	MEDIUM (M)	**>L,**	**<=**	66	minus (result "USER RELIABILITY")/3
	HIGH (H)	**>M,**	**<=**	100	

11.5 Appendix D5 - DOC4 - IT Characteristics

A	B	C	D	H	I	J	K	M	N	O	P
Company	Area	Sub-system	*Application Acr.*	*Devel.Policy 1=internal 2=Package 3=ThirdParty 4=Outsourcing*	*App.Type 1=C/S, 2=Host, 3=PC*	*replacement before Euro advent. (" "= no)*	*Pgm TP*	*Pgm batch*	*Total pgm*	*Maps*	*jcl*
			Totals								

Q	R	S	T	U	V	W	X	Y	Z	AA	AB
Vsam	*DB2*	*DL1*	*Other*	*Tot. arc.*	*Num. prosp./ mod/ print*	Tech/ funct. Adeq.	Organ. Impact	Structural criticality	Mainten. Complexity	Document. Level	*Nr. of Interf. Flows*

11.6 Appendix D6 – DOC5 – Modification / Solution Table

Modification \ Solution	Multiple currency	Duplicate application	Dual currency	Dual amounts	Mask Nat. Currency	Mask EURO	Single curr. EURO
decimalization	yes	yes	yes	yes	no	yes	yes
dual-currency management	no	no	no	yes	no	no	no
currency code management	yes	no	yes	no	no	no	no
change of comput. algorithms	yes	yes	yes	yes	no	yes	yes
modification of constants	yes	yes	yes	yes	no	yes	yes
new functions	yes	opt	yes	yes	no	no	no
output in dual-amount	opt	opt	opt	yes	opt	opt	opt
converter to interface other applications	opt	opt	opt	yes	opt	opt	opt
converter to interface archives	yes	no	yes	no	no	opt	opt
conterter for TP inputs	no	no	no	yes	yes	yes	no
converter for prints	opt	opt	opt	opt	opt	opt	opt

Legend

yes: modification required

no: modification not required

opt: the modification is optional.
Its execution depends on the specific application, on the normative constraints and/or on strategic needs of the company

11.7 Appendix D7 - DOC6 - Euro Solution Characteristics

A	B	C	D	E	F	I	J	K	L	M	N	O	P
Appl. Acron.	*Description*	*Curr. Code*	*Decimals*	*Multi-curr.*	*Appl Euro Compl*	*Limit deployment date*	*Solution type*	*Impact*	*Notes*	*Decimaliz 0=no 1=y (no ark) 2=y + ark*	*Dual amount 0=no 1=y*	*Curr. Code 0=no 1=y*	*Algorit 0=no 1=simple 2=compl*

Q	R	S	T	U	V	W	X	Y	Z	AA
Costants & represent. 0=no 1=y	*New funct. 0=no n=n°funct.*	*Dual amount output 0=no n=n°funct*	*Interface Converter 0=no 1=simple 2=complex*	*Arch. Convert 0=no 1=simple 2=complex*	*TP input Convert 0=no 1=simple 2=compl*	*Prints Convert 0=no n=n°fun*	*Other modifications 0=no n=n°fun*	*Cost for package or service (annual fee)*	*Annual fee for package*	*Cost for Euro adaptation by the provider*

11.8 Appendix D8 – DOC7 – Cost Item / Modification Parameters

	TYPE OF MODIFICATION REQUIRED											
	decimaliz'n 0=no 1=y (no archive) 2=y (with arch.)	*dual currency. 0=no 1=y*	currency code 0=no 1=y	algorithms 0=no 1=simple 2=complex	*constants & representations 0=no 1=y*	New functions 0=no n=nr.functs	*dual currency output 0=no n=nr. functs*	*Interface Converters 0=no 1=simple 2=complex*	*Archive Converters 0=no 1=simple 2=complex*	*Input Converters 0=no 1=simple 2=complex*	*Print Converters 0=no n=nr.functs*	Other changes 0=no n=nr.functs
TP Programs	70%	70%	70%	30%	50%	10%	10%	10%		20%		10%
Batch programs	70%	70%	70%	50%	30%	10%	10%	10%				10%
Maps	70%	70%	70%		75%	10%	10%			20%		10%
Archives									70%			
Interfaces								100%				

11.9 Appendix D9 – DOC8 – Effort Parameters

	Parameter	p/d	Notes
	Person/day cost		***market cost (average)***
1	***effort required for the adjustment***		*To be applied for decimalisation, dual-currency,*
	for each prog. and map		*Currency_Code, Algorithms and Constants*
	p/d to modify a map on host	0.3	
	p/d to modify a C/S map	0.2	
	p/d to modify map on PC	0.1	
	p/d to modify a program on host	1.8	
	p/d to modify a C/S program	1.3	
	p/d to modify a PC program	1	
2	***adjustmets due to special***		
	kind of modifications		
	simple archives (Type 1)	0	
	complex archives -VSAM- (Type 2)	5	add 5 p/d for the adaption of each archives
	simple algorithms (Type 1)	0	
	complex algorithms (Type 2)	p/d * 20%	add 20% to the total estimated p/d (Parameter 1)
3	***Additional effort for CONVERTERS***		To be applied to Interfaces, Archives, Prog_TP
			Prints
	Simple Converters (Type 1)	2	
	Complex Comterters (Type 2)	10	
4	***Standard values for new developments***		To be applied as for new functions
	new function on Host	7	
	new function on C/S	4	
	new function on PC	2	

11.10 Appendix D10 – DOC9 – Other Costs

Parameter	Value	% Cost Increase for implementation	% Other Costs for EDP personnel	% Functional Analysis Costs	TOTAL Other Costs Increase %
	High	7		7	14
structural criticality	Medium	3		3	6
	Low	0		0	0
	High	10			10
maint. complex.	Medium	5			5
	Low	0			0
	High		10		10
organiz. impact	Medium		5		5
	Low		0		0
	High	0		0	0
docum. availability	Medium	3		3	6
	Low	7		7	14
	High	0		0	0
tech/func. adequacy	Medium	3		3	6
	Low	7		7	14
	Min. Value				5%
TOTALS					
	Max. Value				40%

11.11 Appendix D11 – DOC10 – Estimate

A	B	C	D	E	F	G	H	I	J
Area	*Application*	*Acronym*	*Platform*	*Estimate*	*Implement Effort (p/d)*	*Other effort (p/d)*	*Total Effort (p/d)*	*Limit date*	*Notes*

12. Appendix E

12.1 Brief overview of the Renaissance Method.

Many companies today find themselves relying on legacy systems which are problematic but critical to their business. These systems often use old technology and need to be re-engineered to bring them up to date on modern technological platforms so future evolution becomes part of normal maintenance effort.

When faced with reengineering a legacy system, companies are also faced with a bewildering variety of scenarios.

- we have a system which is actually constraining us from increasing market share, what on earth are we to do with it.
- management has told us we must install a modern client server system, how on earth do we transform our current system.

The important point to note is that these systems are critical to the companies employing them and must remain in operation while the evolution takes place. Evolution can take the form of total replacement at one end of the scale to no change at the other to some form of reengineering in the middle.

The Renaissance project devised a method which could be used to assist a company to assess what type of reengineering it needed and then enable it to plan and execute a controlled evolution project. A real process broken down into tasks was devised. These tasks fell into four main activities as shown in Figure 6.

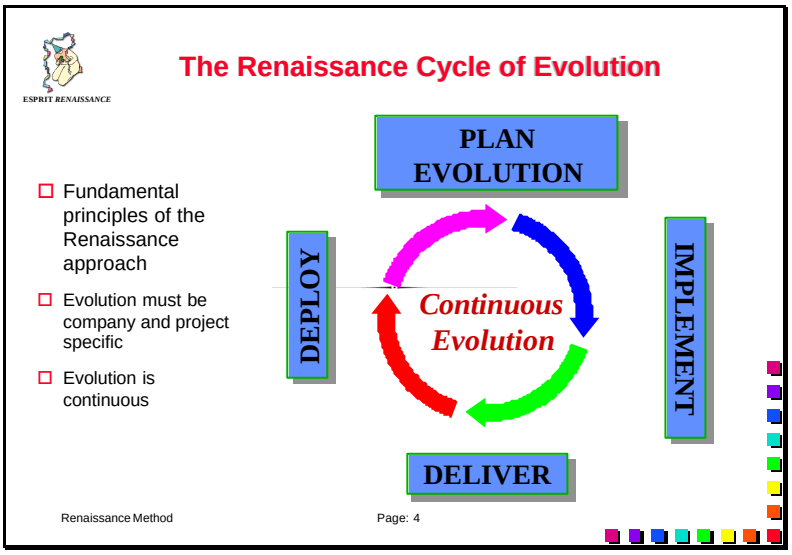


Figure 6 The Renaissance Method

The individual tasks were defined in terms of

- Actions
- Inputs and Outputs
- Responsibilities

The fundamental principles of the Renaissance approach are that

- Evolution must be continuous
- Evolution must be company and project specific

The method devised does not prescribe any particular technique or tool. It provides a framework which can be customised by any user to tailor it to a particular environment and evolution problem.

It caters for the concept of a third party contractor by assuming the parties interested in a project are:

- *Customers* - those who commission the work
- *Suppliers* - those who create the product
- *Sub-contractors* - those who provide products or services to the Supplier
- *Users* - those who operate the product

Often the problem facing a company with reengineering is 'how to start' an evolution project. Given a starting point they can progress under their own steam. Reengineering is expensive (labour intensive, disrupts normal working procedures) and sometimes dangerous (reengineer the wrong parts of a system, miscalculate cost, risk, timescales). The key to successful evolution is a good project plan; the key to a good project plan is a realistic estimate of effort, resources and risk; the key to a good estimate is accurate information; the key to accurate information is thorough assessment. Renaissance encourages a user of the method to start by looking at a system at a high level of detail and follow the evolution planning activities iteratively, including more detail on each pass, until a good understanding of the system and its evolution requirements is obtained. It recommends three levels:

- System – where the system is considered one autonomous unit
- Component - often subsystems, but more often the individual programs and files it uses
- Constituent - the constituents from which the components are built

Determining the assessment level for any iteration can be thought of as a review of the elements of the current system to identify components and characteristics, considered important to the organisation performing the assessment, which can assist the selected activities. Bearing this in mind the first activity of the method is to concentrate the user on the task of collecting information about the system to be reengineered. This startup is effectively organising the foundations of a catalogue of information which can be enhanced with time and used to assess the system. This catalogue is termed the Information Repository in Renaissance.

In addition to technical information, Renaissance encourages a user to describe its Business Processes so that the requirements of these can be addressed.

To prepare for assessment, you gather up documentation, assign assessment staff, organise personnel for interviews, etc. So the method helps you to organise yourself quickly for assessment with the minimum pre-requisites. These are -

- A *customised method*, for your organisation. For example, the responsibilities defined by the method mapped in a sensible way onto your own organisation personnel. Maybe you'll leave out steps you consider unnecessary.
- *Problem statement*. Why is the legacy system "a problem". Are the users unsatisfied? Can your system still inter-operate with others? Maybe upper management has imposed a new company-wide architectural platform. Your solution should address these.
- *Your business goals and constraints*. Maybe you're pursuing market share, or require improved quality / performance / capacity, or maybe you want to introduce new services. Your solution should address these.
- *Your Current System*. This is essentially some sort of baseline "system as it stands now" at the level chosen. This information will be enhanced on each iteration of the planning activities.

Having defined these, you are ready to begin the Evolution Planning phase, the first activity of which is assessment of the current system, at the level chosen for this iteration. Assessment consists of 2 parallel activities:

- Model context of current system –

to provide clarification of details for the assessment by illustrating how system components interact.

- Assess current situation –

  To assess whether the system is a candidate for evolution and provide information to support the selection of an Evolution Strategy and Estimating activities. It involves:

- Decide on technique for making assessment

  This can range from expert opinion through to formal metrics.

- Decide what aspects of the system, at the level previously specified, to use in the assessment.

  A system can be thought of as a hierarchy of interacting elements. For the purpose of system assessment within Renaissance these elements can be organisational factors, hardware, support software and application software. Each element type consists of components possessing characteristics which can be measured.

- Gather information to inform this process
- Do the assessment

  In addition to providing information for estimating cost and risk, an assessment is carried out to decide whether or not your systems actually need reengineering and to determine priorities. For this reason Renaissance uses the concept of Portfolio Analysis. You can say that those systems of a high Business Value but of a low Technical Quality are the systems most in need of reengineering. Plotting BV against TQ means that those systems, which fall into the bottom right hand quadrant, are good reengineering candidates. This is not a recipe for assessment, just general advice. Assessment can be expensive so this method narrows focus to those systems most in need of re-engineering

  Assessment involves calculating:

- Technical Quality of a system.

  This involves assessing the characteristics chosen using the chosen techniques and aggregating the results. To assess these Renaissance recommends scoring each selected characteristic on a 4 point scale so that an objective picture of the system can be built up. Generally you can say a score < 3 means some action is needed.

- Business Value

  This involves a process known a Business Value Analysis.

All assessment and modelling information should be catalogued in the repository and, as the method progresses, the information collected will clarify the understanding of the system and its components' relationships.

Having assessed the system to see if it requires reengineering, you now need to determine which is the best way to transform it. There are any number of evolution strategies for software reengineering but Renaissance uses 4 main strategies:

- *Revamp* – where the user interface only is modernised.
- *Restructure* – where the user interface is unaltered but the internal structure of the system is improved.
- *Redesign with reuse* – where the system is rewritten using much of the original code.
- *Rearchitect* – where the system is re-implemented on a completely new technological platform.

Having assessed the current system and decided on maybe several possible evolution strategies and target systems, a user will need to model these and estimate size, cost and risk and possibly do a Cost Risk Analysis to determine the optimum strategy. Having determined this for the individual system components, an overall evolution strategy must be formulated. In this task the individual evolution strategies, together with the individual components of the

current system on which they are to be applied, are organised and documented in the overall system level evolution strategy. This will serve as a roadmap for the developers of the evolution project plan in the next phase of the method where the transformations corresponding to the implementations of the individual strategies will be planned and executed.

The first iteration at a high level gives an idea of the state of the system and allows you to prioritise its reengineering needs along with those of other systems in your organisation. You repeat the method at the next level with more detail until you have sufficient detail of the new system to enable you to formulate a sound project plan.

Eventually, the results of evolution planning enable you to make a **Go/NoGo decision** because:

- With the business goals in mind, you will have assessed the technical quality and business value of the system.
- You will have developed a set of candidate evolution strategies taking into account
- constraints,
- the problem statement,
- business goals
- You have a cost/benefit analysis of each candidate
- You have identified and costed an overall system evolution strategy

For a 'go' decision, the next activity is the Implementation activity. The first task for implementation is Project Planning. This is the classical project planning activity which maps estimated activities and costs to a formal project schedule and budget.

An evolution project is obviously different from a traditional implementation project due to the fact that there is already a system in place. Even though you have already *selected* your evolution strategies, you will need to understand the system in more detail in order to *implement* them, that is, do your detailed system design. The implementation phase deals with the actual System Transformation of the legacy system to the desired target system. This Transformation must be *controlled* so Renaissance recommends the identification of *Component batches* which can be migrated together. This enables the evolution process to be one of continuous, incremental testing and transformation. The Implementation Phase in Renaissance does not consider the target environment for the system

- implementation may be in-house with third party suppliers and sub-contractors, not working on customer site
- the customer may not even be involved during implementation.

On completion of Implementation, we have the Delivery activity. While implementation may be done in isolation, Delivery will very definitely involve the customer. It involves the acquisition and on-site installation of new hardware and software packages along with any infrastructure needs; the identification of acceptance criteria in order to validate the target system; preparation for, and onsite acceptance testing; separation of concerns from implementation team and operational team where we must look forward to and plan for deployment and, finally, data migration. Data migration from a legacy system to new system can be complex and expensive and frequently the problems of this task are overlooked and underestimated.

Which brings us to the final activity, Deployment. One of the most neglected aspects of deployment is how fast the organisation should switch over from the old system to the new system. Gradual? Abrupt? Co-existence for a time? Deployment is how to control the changeover process and can vary considerably according to the nature of the reengineering project. For example, a restructured version of a legacy system will not have the same impact on working practices as a re-architectured one will. During deployment, training programs will be necessary. These may be customised according to the degree of transformation that occurred

Earlier in this section we said:

Evolution must be continuous. Renaissance achieves this by a process of *in-use evaluation* which identifies criteria for future smooth evolution of the system through monitoring critical factors such as:

- User satisfaction with system functionality
- System performance (response, number of users, etc.)
- Evolving business goals of the organisation, and contribution of the system to these goals
- Documenting the revised business process supported by the new system

New requirements arriving from business community and user community can be ploughed back into the method for the next round of reengineering.

Evolution must be project specific Renaissance achieves this by allowing project constraints to be input during method startup

- "Management has mandated this particular platform."
- "We only need top-level assessment, nothing detailed. Our budget is too small for anything else."
- "This is only a feasibility study. We only need the Evolution Planning phase."
- "We already know exactly what we have to do. We only need the Implementation, Delivery, and Deployment phases."

Evolution must be company specific Renaissance achieves this by allowing company tools and techniques to be specified for the Renaissance activities

- "We have a tried and true method for risk management that is used organisation-wide."
- "We have standardised on this software for project management."
- "We use the XYZ design documentation method."

"We have bought the toolset from Company X for reverse engineering support."

12.2 The Renaissance Method Activities and Tasks

Below is a list of the sections in the Renaissance Method document which handle the actions performed during the execution of the method. The numbers correspond the section numbers in the document itself.

2.3 Plan Evolution

2.3.1 Startup Method

2.3.1.1 Create Repository Baseline

2.3.1.2 Identify Constraints and Business Goals

2.3.2 Describe Business Process

2.3.3 Assess Current Situation

2.3.3.1 Specify Assessment Levels

2.3.3.2 Select Assessment Characteristics

2.3.3.3 Assess Characteristic I

2.3.3.4 Global Assessment

2.3.4 Model Context of Current System

2.3.5 Select Target

2.3.5.1 Elaborate Possible strategies

2.3.5.2 Assess Strategies

2.3.5.3 Choose Strategy

2.3.5.3.1 Size and Cost Estimation

2.3.5.3.2 Risk Assessment

2.3.5.3.3 Cost Benefit Analysis

2.3.5.3.4 Develop Overall Strategy

2.3.6 Model Context of Target System

2.4 Implement
- 2.4.1 Plan Evolution Project
- 2.4.2 Define Test Strategy
- 2.4.3 Design Target
  - 2.4.3.1 Model Current System
  - 2.4.3.2 Model Target System
- 2.4.4 Design Transformation
- 2.4.5 Transform
- 2.4.6 Test
  - 2.4.6.1 Plan Test
  - 2.4.6.2 Prepare Test
  - 2.4.6.3 Perform Test
- 2.4.7 Prepare Target

2.5 Deliver
- 2.5.1 Plan Deployment
- 2.5.2 Install System
- 2.5.3 Migrate Data
- 2.5.4 Do Acceptance Test

2.6 Deploy
- 2.6.1 Design Changeover
- 2.6.2 Train Operators
- 2.6.3 Replace Old System
- 2.6.4 Document Revised Business Process
- 2.6.5 Do In-Use Evaluation

12.3 Responsibilities in the Renaissance Method

These are not one to one correspondences with human beings. They describe 'capabilities' that are involved in the application of the method. The actual correspondence between these and the members of an evolution team are specific to the evolution project and organisation performing it. Below is a list of responsibilities and where they are required in the method.

Responsibility	Acronym	Where used
Application Business Expert	ABE	232, 2334, 234, 236, 2432, 264, 265
Client Representative	CL	236
Data Modeller	DM	
Evolution Modeller	EM	2311, 2312, 2331, 2332, 2333, 2351, 2352, 23531, 23534, 244
Estimator	EST	23531, 23532, 23533
Legacy Functional Expert	LFE	234, 2431
Legacy Implementation Expert	LIE	234, 2431
Legacy System Operation Manager	OPMG	2312, 2332, 2352
Target System Operational Technical Service	OTS	247, 251, 261, 262, 263, 264, 265
Quality Engineer	QE	23532, 23533, 242, 2461, 2462, 2463, 254
Software Architecture Expert	SAE	2432

Software Engineer	SWE	2311, 232, 2331, 2332, 2334, 234 2352, 23534, 236, 2431, 2432, 244, 245, 2461, 2462, 2463, 247, 251, 252, 253, 261, 262, 263, 265
Software Project Manager	SWPM	2311, 2312, 241, 245, 251
Test Engineer	TE	242, 2462, 2463, 254
User	USER	2312, 2332, 234, 2352, 236, 242, 2432, 2461, 2462, 251, 252, 253, 254, 261, 262, 263, 264, 265
Additional for emergency		
Euro Expert	EE	Yet to be decided

Table 7- Responsibilities in the Renaissance Method

12.4 Renaissance Tasks and Responsibilities

Activity	Responsibilities
2.3.1.1 Create Repository Baseline	EM, SWE, SWPM
2.3.1.2 Identify Constraints and Business Goals	EM, OPMG, SWPM, USER
2.3.2 Describe Business Process	ABE, SWE
2.3.3.1 Specify Assessment Levels	EM, SWE
2.3.3.2 Select Assessment Characteristics	EM, SWE, OPMG, USER
2.3.3.3 Assess Characteristic I	EM
2.3.3.4 Global Assessment	ABE, SWE
2.3.4 Model Context of Current System	ABE, LFE, LIE, SWE, USER
2.3.5.1 Elaborate Possible strategies	EM
2.3.5.2 Assess Strategies	EM, OPMG, SWE, USER
2.3.5.3.1 Size and Cost Estimation	EM, EST
2.3.5.3.2 Risk Assessment	EST, QE
2.3.5.3.3 Cost Benefit Analysis	EST, QE
2.3.5.3.4 Develop Overall Strategy	EM, SWE
2.3.6 Model Context of Target System	ABE, CL, SWE, USER
2.4.1 Plan Evolution Project	SWPM
2.4.2 Define Test Strategy	QE, TE, USER
2.4.3.1 Model Current System	LFE, LIE, SWE
2.4.3.2 Model Target System	ABE, SAE, SWE, USER
2.4.4 Design Transformation	EM, SWE
2.4.5 Transform	SWE, SWPM
2.4.6.1 Plan Test	QE, SWE, USER
2.4.6.2 Prepare Test	QE, SWE, TE, USER
2.4.6.3 Perform Test	QE, SWE, TE

2.4.7 Prepare Target	OTS, SWE
2.5.1 Plan Deployment	OTS, SWE, SWPM, USER
2.5.2 Install System	SWE, USER
2.5.3 Migrate Data	SWE, USER
2.5.4 Do Acceptance Test	QE, TE, USER
2.6.1 Design Changeover	OTS, SWE, USER
2.6.2 Train Operators	OTS, SWE, USER
2.6.3 Replace Old System	OTS, SWE, USER
2.6.4 Document Revised Business Process	ABE, OTS, USER
2.6.5 Do In-Use Evaluation	ABE, OTS, SWE, USER

Table 8 – Renaissance Tasks and Responsibilities