

## 3. Business Preparation

### 3.1. Business Process Modelling and Target States

A good starting point for preparing for a COTS transformation project is business process modelling. This involves identifying and mapping the operational processes of the organisation. In preparation for COTS implementation, each business process needs to draft their target states (the desired outcomes of the system), which form the basis for the more detailed descriptions of the functionalities. Target states can be fairly simple documents describing what the targeted changes to the process are, what they will achieve and what will remain the same.

It is important to make decisions about the target states, including intended organisational changes and/or changes to roles and responsibilities, before the COTS implementation work starts. If these decisions are not taken early in the implementation process, they may cause additional problems and delays later on.

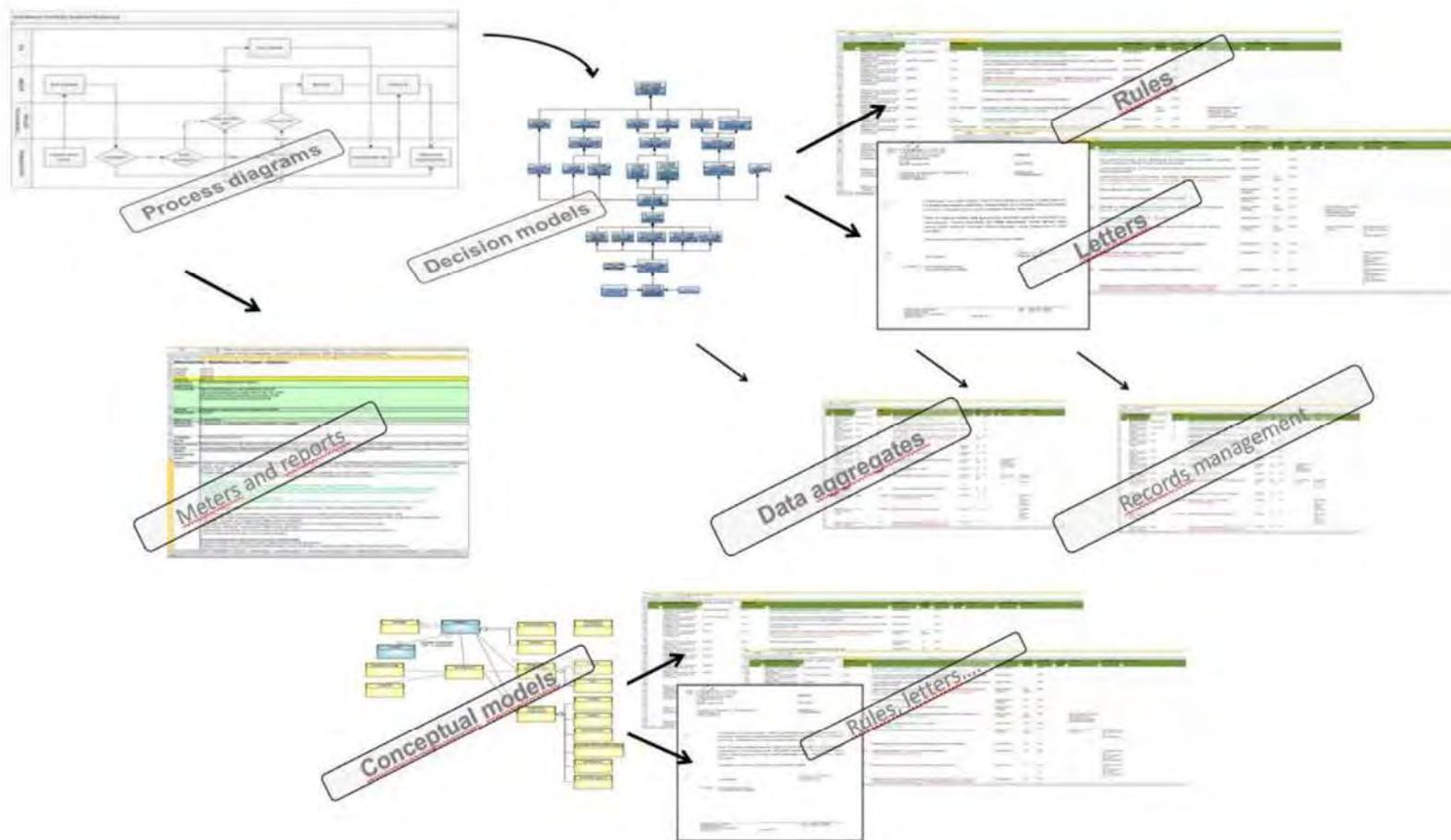
### 3.2. Business Rules

In the preparation phase, functionalities should be described in as much detail as possible. The setting of detailed requirements, as are usually done for traditional software development, are not generally a good way to proceed with a COTS product. This can lead to the production of rigid requirements that attempt to replicate the legacy systems. This can in turn lead to excessive customisation and the loss of much of the standardisation benefits of an off-the-self product. It is more advisable to focus on the outcome, what is needed and why, and to describe this in business rules. The golden rule is that with a COTS product, care should be taken to avoid describing how things should be done.

Business rules can be related to any functionality such as tax or interest calculation, risk management or processing of data. Business rules can be found in the documentation of the legacy systems, user instructions or sometimes directly from the legislation. However, it is highly likely that not all the necessary rules will be documented. Some might exist only in the legacy system code. The more that areas without sufficient documentation can be identified before the implementation starts, the easier the implementation will be, since the content of the missing rules can be decided and documented as part of the preparation. Even then, some of the missing rules may only be noticed during the implementation phase, and this may require fast decision making from the business personnel in the project.

The business rules should be documented and classified based on their purpose. It should also be clear which data elements are necessary for the rules to function properly. One recommended way of classifying the business rules is decision modelling. Decision modelling helps to fill out the gap between process models and business rules. It enables modelling of the business by dividing it into concrete parts that are understandable to business people without being too detailed. It also helps not to lose sight of the overall picture of the process while delving deep into the detailed rules. Decision modelling can be done with most of the applications used for modelling, as the notation is usually straightforward.

Figure 3.1. Business preparation



Source: Finnish Tax Administration (2019)

When writing the business rules, there should be a good understanding of what is actually required by the legislation and what is the tax administration's interpretation (which may go beyond minimum legislative requirements). The writers should keep an open mind, question the way things have been done before and try to avoid defaulting to the way things were done in the legacy systems. They should also try to find ways to simplify whenever possible, since complications will increase the work needed for development and testing.

In addition to written rules, it can be helpful to use charts, diagrams, pictures and examples to make things more understandable. The rules should use coherent terminology with acronyms, abbreviation and jargon explained. Business rules should be reviewed widely to ensure that everything is included and to help everyone understand what is being done.

In case of disagreements or perceived unreasonable demands from any of the business areas, the team responsible for business preparation should be able to refer matters to a senior official person with the power to simplify the rules or, in some rare cases, discard the rules entirely. This might be done in extreme cases by escalating the questions or issues into the steering group of the transformation project.

### 3.3. Data Modelling and Records Management

The conceptual data model collects the information needed for the process, i.e. the characteristics of the data, and groups them into concepts and describes their relationship to each other. Concepts and attributes are collected from the information that the process receives, uses, and produces. The concept work also collects the codes and code values that the business processes need.

In Finland's case, State offices in Finland are required to define retention periods and other archiving rules for the records they process. The rules should also contain descriptions of any restrictions on access to records, including confidentiality and data protection requirements. In EU Member States, regulations regarding individuals' data rights (General Data Protection Regulation EU 2016/679) must also be taken fully into account regarding archiving requirements, including requiring records and other data to be removed from the systems when it is no longer needed.

The retention rules (including the number of years data should be retained and the retention start date) are defined by each business process area. This ensures that the operational needs for records are taken into account. This predefined metadata and the operational metadata (e.g. the received date of the data) can be used to create reports and help the organisation to become aware of the different attributes of the data in its possession.

After the operational retention period has reached its end, the data can be moved from the standard production store into an archiving store. After the passage of any required retention period following archiving, the archived data can be purged (i.e. permanently deleted).

### 3.4. Letters and Print-outs

There is a global and growing trend towards the use of plain language in public administration. A COTS project provides a great opportunity to review and potentially to rewrite all the letters and decisions produced by a tax administration in plain language to achieve multiple goals. Tax administration's letters should be easy to understand for everyone, not just specialists. If the letters and instructions are not understood, customers may make mistakes or may contact the tax administration to ask for clarification, which can place burdens on taxpayers and the tax administration. Poor formatting and unclear writing styles can also give an unprofessional appearance, impacting the reputation of the tax administration.

The goal should be to write clear, understandable and professional letters to taxpayers. These are some useful guidelines for the writing style to be built into COTS, including:

- Consistency in language, style and layout
- Informative headings
- A clear explanation of the reason the letter is being sent
- Highlighting the most important information at the beginning of the letter
- Giving proactive guidance where more difficult issues are covered
- Clarity as to requested actions and directions
- The avoidance of “legalese” or jargon
- Addressing the reader directly and avoiding use of the passive voice or third person
- A layout which helps the reader navigate the letter at a glance.

In addition to creating guidelines to help the composition of letters, it helps to create letter templates for different types of letters (decision letters, requests for information, appeal decision letters etc.). Each area/process should then make use of these templates and produce a list of all their letters (a letter inventory); word templates of all their letters, including all possible standard letter content/paragraphs; and letter rules, describing the data that is shown in the letter. It is good practice to involve both taxation and language experts in drafting the word templates and, where possible, experts in behavioural insights.

### 3.5. Data Cleansing

Typically there is a need to transfer data from the legacy applications or databases to the COTS system. This data may be both master data that operational processes utilise (e.g. customer data) and operational data (e.g. returns or appeals data) that will be further processed in the COTS systems. Tax administrations may think that their taxation data is in good shape, but often on closer examination there will be quality issues which need to be addressed, particularly in older data.

Data cleansing is the process of detecting and correcting/removing corrupt or inaccurate records from databases or other data sources. It is highly recommended that data cleansing is done before the data is transferred to the COTS system. If this is not possible, senior management should be made aware of the impurities the data may contain and of the issues that may cause. Data cleansing is not usually possible during the conversion process, because it influences conversion reconciliation.

Work on identifying any issues with data quality should be started well before the implementation project starts. The first task is to identify, on a high-level what data, and from which data stores, needs to be transferred to the COTS system. The second task is to analyse the quality of the data that should be transferred: i.e. whether the data is complete, correct, and accurate and does not contain irrelevant information. As a third task, data cleansing activities should be commenced, starting from the most business critical data.

Where errors or inaccuracies are systematic and the volumes are high, cleansing can be done through automatic programmes. Smaller volumes or difficult cases can be corrected manually. Both methods are labour intensive and affect COTS transformation costs and the schedule. The data quality must be taken into account when planning the COTS transformation scope, schedule and costs. The worse the data is, the more time and resources will be needed for data cleansing.

Data quality will also affect decisions regarding the decommissioning of legacy applications. If the data cannot be transferred from legacy systems to the COTS system for quality reasons, then legacy systems will need to be maintained and possibly complex integration work and manual interventions may be needed.

### 3.6. Transition Period Planning

The transition period refers to the period of time when business processes are in transition from the legacy systems to the COTS solution. Transition period planning should focus on comparing the target state and present situation and drawing a map on how to get to the target state. Transition period planning is cross-functional and combines issues around the business, legislation and IT, but it should be driven primarily from the business point of view.

Transition period plans should include the following: changes in taxation; the impact on processes; staffing changes; changes for the customers; changes in the required or provided data; and changes in interfaces. It should describe what happens after the implementation project or each part of the implementation project.

A problem that can occur in transition period planning is that not enough is known about the change required or the implementation method. It is recommended to have some COTS experts to support the business in doing the transition period planning.

### 3.7. Change Management

The COTS transformation project is likely to affect most employees of a tax administration to a greater or lesser degree. The importance of change management cannot be underestimated. People will often not just be affected by the new COTS tool, but also by changes in legislation, work procedures and responsibilities. It is a fact of most COTS projects that staff will have to face long periods of uncertainty and constant change. It is also very likely that not everyone will welcome the changes, at least not immediately. For this reason, it is recommended to invest heavily in building the resilience of staff during the project, including through effective communication, to ensure that staff maintain focus and energy over the whole COTS transformation.

Change management, in particular taking the people aspect fully into account, has to be an integral part of project planning. It is not something that can be separated from the project work. The main aim of change management is to help people to accept and adapt to the change. They need to understand the real effects of the change on their everyday work, on the achievement of the organisation's goals and also the need for, and necessity of, their own personal professional development in making the change a success.

There are many types of theoretical frameworks for change management, but changes resulting from an extensive COTS project are far from being linear, so no step-by-step system or methodology can be used as such. In addition, tax administrations, even during a large COTS implementation, need to adapt to other complex and rapid changes in the operational environment.

In any transformation, top-management commitment is of crucial importance. They have to be onboard from the very beginning, and to stay there. This is why an open and constant co-operation with business unit directors is critical. They need the tools and information to show their commitment to their staff, to create and maintain a future-oriented culture and to repeat the "why" in their communication, especially when challenging times occur.

Senior directors should also be included in the steering group of the transformation project and therefore be involved in the decision making at every step of the transformation project. This involvement should help to keep them more committed to the transformation even in a long project.

Another key group is the supervisors of the employees. It is primarily their task to make sure that the employees' expertise and skills are on a sufficient level to be able to perform the tasks required. Supervisors have to encourage and motivate employees to learn new skills. It must be noted, however, that supervisors often have a dual role. In addition to being promoters of change, they will also need to learn new skills. Constant, transparent and two-way communication between the supervisors and the COTS project team needs to be established. In Tax Finland, the COTS change management team includes a representative from each business unit. This creates an open channel and a "help-desk" for the supervisors. This is a channel for communication with employees in other business roles, as well.

Communication is an integral element of change management. Both internal and external stakeholders should be identified and their needs analysed. A communication plan should be created based on the stakeholder plan. The communication plan, and the actions included, needs to be constantly reviewed and updated, as the situation may change as things proceed and sometimes quite rapidly. It is important to be transparent and open in communication. If communication takes place too late, or if things are not communicated at all, rumors can abound and flourish.

Even though a lot of communication takes place electronically, for example through an intranet, bulletin board or e-mail, the importance of face-to-face interaction needs to be remembered. It is a great way of showing appreciation and commitment as well as gathering feedback.

## 4. Implementation

### 4.1. Methods and Tools

COTS vendors and tax administrations usually have their own methods for software implementation. These methods might contain many of the same elements, but there might also be differences. It is recommended to use the methods of the COTS vendor as far as possible, since they should have experience of successful COTS implementation projects and know which methods have worked best from that perspective. It is important to ensure that the project staff is trained to work according to the vendor's methods.

COTS vendors might not be that focused on the change management issues. Tax administrations should make sure that change management processes are also in place. In addition, production support and maintenance after the go-live events should be planned and agreed upon. Typical tools to help the project governance and day-to-day work used in a COTS implementation include:

- Project planning and management tools for project tasks, resourcing and resource assignment, scheduling, total work and cost estimation, and follow-up
- Document management tools e.g. for document storage, approval and communication
- Testing management tools
- System configuration management tools.

It is important that the project staff has access to these tools, that the tools are easy to use and that support is available.

It is also important that the COTS vendor has appropriate automated tools. Tools for continuous integration management, configuration version management, release management and change migration between environments are especially important in a COTS project. The COTS vendor should be able to quickly and precisely make the desired changes to any environment. Use of manual rather automated tools in change migration is usually the reason why something that works in a test environment does not work in production.

### 4.2. Requirements

Defining the detailed requirements for the system is a crucial part of the implementation project. Making sure that everyone understands the requirements the same way takes a lot of communication and is a prerequisite for obtaining the kind of system that the business needs.

The first phase should be to create a task plan to go through the requirements. All the high-level requirements should be included in the task plan and the definition order should be logical. The task plan should be at a sufficiently detailed level to ensure that nothing important is left out and that the most difficult things are addressed early enough in the process. It should also be flexible enough to allow for possible changes, both to the tasks and the schedule. All parties should be involved in approving the task plan.

Requirement meetings should be held according to the task plan to go through the detailed requirements. A detailed agenda for every meeting should be available well in advance to enable

proper preparation for the meeting. Meetings should be interactive fora where ideas are shared and the rationale behind proposed requirements can be explored in depth. It may not be possible to go through every detailed rule, but at the very least the most important ones should always be discussed and agreed upon.

Everyone in the meetings should be open to discussing new ideas and willing to think of alternatives and compromises. During the requirement meetings, many quick and bold decisions are needed. This can be stressful and the decision making of individual business process owners should be supported by senior staff in the project and, if necessary, outside of the project as well.

After the meeting, minutes should be written and reviewed, so that everything important has been included and everyone understands what has been agreed upon. There should be an agreed deadline in place for writing and reviewing the meeting minutes. Meeting minutes should also include the business and calculation rules (or at least links to the appropriate documents) that have been discussed and agreed upon. Meeting minutes should then be clearly linked to the relevant system development tasks and test scenario groups, for requirements traceability.

Communication should naturally not be limited to meetings. On the contrary, constant discussions should be encouraged between developers and subject matter experts (SMEs). It is important to maintain the discipline, though, that decisions made outside the meetings should also be recorded. After the go-live, there should be a documentation review in place, to make sure that the documentation includes the entire configuration, is up to date and of good quality.

With a COTS product, the subject matter experts should be able to see the system configuration soon after requirement meetings. This enables early review of the configuration. This is important for pre-testing as to whether the most important things work as intended and that there have not been any misunderstandings.

Informal demos are a good way to present the developed functionalities. Even when this pre-testing is informal and does not cause any contractual obligations, it is advisable to have it as an expectation for the SMEs to document that it has been done. Pre-testing prevents a lot of unnecessary work and delays in the actual testing phase. It also helps in keeping track of what has already been developed and is ready to be tested.

One thing to keep in mind is that there are many similarities between different functional areas and it is beneficial to solve the same issues in the same way in every area. This makes the maintenance easier and improves usability of the systems and transferability of skills. Vendors do not necessarily pay too much attention to this and even if they do, it still takes a lot of effort and specifically assigned personnel from the tax administration to ensure this kind of approach is taken.

### 4.3. Data Conversion

#### *Definition and Goals*

Data conversion is the process by which data from one or more legacy software applications is made useable in a new system in which the business functions operate.

1. In conversion work, it is vital to clarify the following topics:
  - Scope: What data is needed in the future and from which period
  - Data inventory: Where is the data stored in the legacy system or other store
  - What data will be converted automatically and what will be converted manually
  - Data cleansing: Where does responsibility lie for fixing bad data before conversion

- What data is needed in the operational database and what data can be stored in the data warehouse for read-only purposes?

### *Conversion Roles*

The conversion team should contain people who understand how the COTS product is implemented and how it works, and people who understand how and where the data is located in the legacy systems. Particular roles are:

- Conversion primary subject matter expert who is responsible for the conversion from the tax administration side
- Conversion subject matter experts who:
  - are responsible for conversion requirements and conversion reconciliation in a particular business area
  - participate in conversion verification
  - are familiar with the COTS product functionality
- Conversion Leads and Developers who are:
  - responsible for the conversion from the COTS product supplier side (conduct of the work, documentation, load process and reconciliation)
  - responsible for developing the conversion load process
- Extractor Architect who is responsible for the extraction architecture and documentation
- Extractor Lead who is responsible for managing extraction work and documentation
- Extractor who is responsible for extracting data from the legacy systems
- Technical Expert who is responsible for legacy data knowledge
- Conversion Verification Testing Lead who is responsible for planning verification testing and managing the work of those testing conversion verification and reporting
- Conversion Verification Tester who is responsible for conversion verification testing.

### *Conversion Process*

In the conversion process, it is vital to understand the processes and the data needs, and map the data between legacy systems and the COTS product. It is recommended to identify all the data sources before the COTS project starts. High-level data models of legacy systems and descriptions of databases are also needed. Data cleansing should also start well in advance, particular if the legacy systems are known to contain bad or incorrect data.

Conversion work starts with identifying all the legacy data sources from where the data could be converted to the COTS product. This should be documented in the Data Inventory.

The scope of the conversion work and the requirements will be discussed in meetings of functional areas. It is important that all parties understand the needs and share a common understanding of how data will be mapped between the legacy system and the COTS system. The meeting minutes

will be the base for production of conversion documentation. This documentation should contain the business scope and rules, rules for extracting the data, and how the data is mapped between legacy systems and the COTS product. If the extraction rules are very complicated, it is recommended to define them in a separate document. This documentation will be the basis for testing.

Before the actual conversion during the cutover, conversion should be practiced and tested. This test conversion should be run to a planned schedule, for example every third or second week. It should use the production data to make sure that all the problems with the process or the data will be discovered as early as possible and can be corrected before the actual conversion. These test conversions are called mocks and they can contain part of the customers or the data related to customers (a partial mock) or all of the customers and their data (a full mock). Each mock will increase the amount of functionality until everything is included (a complete full mock). The complete full mocks will also be used in the converted data testing and end-to-end testing.

Manual conversion can be used if there is a small number of cases or the conversion rules are very complex. Extractors will run lists of these cases and the data will be added manually after the cutover.

### ***Conversion Verification and Reconciliation***

Conversion verification testers will verify that the data converted to the COTS product is correct. This will happen by comparing the data in the legacy user interface and the COTS product user interface. In addition, automated testing tools can be used for verifying large amounts of data.

Conversion verification should continue until the cutover period, to make sure no part of the conversion has broken down. There can be errors in extraction, the load process or the functional methods that are used. In addition, the legacy data might need cleansing.

Conversion will be approved following reconciliation and verification. Reconciliation reports should show that legacy, extract and loaded counts and financials are all in balance. Through this report, it is possible to confirm that all the data has been converted successfully.

Conversion reconciliation should be practiced with every full mock so that everyone is familiar with the process and the numbers by the time of cutover.

### ***Conversion Success Factors***

Conversion is a time and money consuming effort. Converted data will impact business development for several years ahead. Key success factors include:

- only converting the data that is really needed. This principle applies especially to historical data. Consider the alternative of using some of the converted data only for viewing and not storing it within the COTS's operational database
- close working within the conversion team between legacy experts, COTS product experts, subject matter experts and testers
- good communication and adequate documentation
- a full understanding of how the COTS product processes work
- mapping the data from the legacy systems to the COTS product
- use of the production data in “mocks” and verifying the results in the reconciliation, conversion verification and converted data testing

- legacy data should be consistent and uniform to make conversion rules as simple as possible. All bad data should be cleansed before data is extracted.

**Box 4.1. Conversion – Statistics for the 4<sup>th</sup> implementation project**

- 6 153 114 Accounts
- 97 426 379 Transactions totalling EUR 21 211 870 000 debits and EUR 21 883 485 000 credits
- 94 702 531 Images

*Source:* Finnish Tax Administration (2019).

## 4.4. Interfaces and Integrations

### *External Integrations*

A modern tax system is highly integrated into various parts of taxpayers' natural systems, including third party systems. Examples can include customer data by customer base registers, incoming and outgoing payments, information on taxable income provided by financial institutions, data on withholding of income tax by employers or information provided through e-invoicing systems or online cash registers.

A large part of a COTS taxation project is replacing the external integrations of the legacy systems and building new integrations as appropriate. As a strategic decision, it might be advisable to keep the external integrations as they are during the COTS project. During the project, there will be many ideas on how to improve and upgrade the various integrations, but getting the project done to time and budget will require keeping tight control of the scope.

It is a highly recommended best practice that all the external interfaces are catalogued. This can help to ensure that all the needed integrations are done in time and it also makes it possible to analyse if there are overlapping or redundant interfaces in the legacy system.

Integrations are usually technical in their nature. The subject matter experts from various business areas are usually able to tell what information they need from each external source or what information they need to provide to an external party. However, they do not usually know the format or the schema in which the information flows. Technically oriented people are needed to specify and test the integrations and to collect all the needed information into a catalogue. Communication with the integration partner is also needed in designing, developing and testing a new or continuing integration in the COTS system. It might even require the revision of contracts with external parties. Some of the integrations transfer massive amounts of data and some of them are part of an online transaction, so planning for performance testing on integrations is necessary.

### *Temporary Internal Integrations*

On top of external integrations, temporary internal integrations are needed. It is not advisable, or usually even possible, to do a taxation COTS project in a single roll-out. During the transition period the COTS system will replace part of the legacy systems but some parts will still remain. At this stage, COTS and legacy systems will usually need to exchange a lot of information and many temporary internal integrations are needed. The aim should be to keep the number of temporary internal integrations to a minimum as they are expensive and error prone. Even when these

integrations are temporary, they are critical to continuing operational reliability during the COTS project. The building of these integrations should be planned very carefully. The needed integrations depend on the legacy system architecture as well as on the COTS structure and logic. Temporary integration challenges can be one of the key design principles in the phasing of the COTS project.

The whole life cycle of the integrations and the changes between roll-outs need to be planned. Usually the main challenges in the temporary integrations are sharing customer data and keeping financials in balance between the two systems. In sharing customer data, a vital choice needs to be made in the beginning as to which system will be the master system for customer data. The master system is the one in which changes are made to customer data. All other systems will only be able to read customer data but they will not be able to amend it.

Tax Finland chose to make the COTS system the master system for customer data in the first roll-out. This decision was taken because it helped in the development of the customer data features. However, a complicated integration had to be built between the COTS and the legacy system. It is worth considering carefully whether it would be preferable to keep the legacy system as the master of customer data until the last roll-out. Either way, this will require very careful planning and testing.

The difficulty of synchronising the financial balances between systems depends on the financial rules in the tax legislation. If all finances are very tax type specific, it can be quite easy. If not, as was the case in Finland, a system needs to be built where payments are used in a predetermined order between systems.

Temporary internal integrations should be as simple as possible. As a general rule, asynchronous integrations are easier to create and maintain and better for detecting errors compared to synchronous integrations.

### *Externally consumable API's*

One of the major challenges for a modern revenue agency is to be able to integrate with numerous third parties to exchange taxation related data. In Finland, a large portion of taxation data is collected from third party data providers such as employers or financial institutions. This enables taxation to be precise and at the same time very easy for the taxpayer. As digitalisation continues to grow rapidly across society, third party data providers are changing. One highly visible phenomenon is the growth of the online platform economy, including the sharing and gig economy. As this becomes a bigger part of the economy, it will be critical to be able to automatically exchange information with platforms. Externally consumable API's seem to be the solution for technically implementing increased data exchange in the future.

Tax Finland's COTS product does not offer out-of-the-box externally consumable API's within its business functionality. It is very likely that such an interface is not possible on a general taxation level. Tax Finland's COTS product does, though, offer the framework capability to design and build the desired business API services and provides the supporting functionality such as a developer portal. Designing and implementing API services in accordance with the agencies' business is seen as normal ongoing implementation and maintenance work in the COTS solution.

## 4.5. Testing

Testing is performed to verify that the COTS product and legacy applications fulfil the business requirements. The amount of work needed for testing should not be underestimated. As many as 50% of the project hours might be spent on testing.

A COTS implementation project includes the same test phases and methods as any other integrated software implementation. These include:

- Unit testing (performed by the COTS vendor) to verify that COTS functionality is ready from the development standpoint and that end-users can start wider system testing
- System testing to ensure that COTS functionality covers business needs and that the COTS system is successfully integrated
- End-to-end testing to ensure that business cycles (e.g. due dates) can be run in the COTS systems and that business processes are supported
- Usability testing by end-users and customers to ensure that the COTS system is as user-friendly and easy to use as possible
- Regression testing to ensure that the new COTS functionalities have not affected already developed and tested functionalities
- Performance testing to ensure that the COTS system can cover known business and data volumes
- Security testing to ensure that user rights and external security requirements are fulfilled
- Technical testing, for example system recovery testing and system security testing.

The testing process includes planning for tests, test scenario writing, test data preparation, test scenario execution and retesting, after defects have been corrected. Test scenario writing and test data preparation take time and may even take more time than the test execution itself. Thus test scenario writing should be started well before the actual test execution starts so that the missing scenarios will not be a bottleneck for progress with the testing process.

It is also important that scenario writing and planning is done by subject matter experts (SMEs) that understand the business needs and are able to verify that testing coverage is on an acceptable level. Testing should not focus only on the simple test scenarios, but also on scenarios that are known to be complicated.

A large scale COTS project may benefit from a separate testing support team including experienced subject matter experts, with appropriate technical knowledge, who are in charge of test planning, coordination (also with external testing parties), testing tools, progress reporting, test environments, test material creation as well as supporting and training less experienced testers for their new job.

The iterative nature of system development means that frequent retesting is needed as changes and corrections are made. This requires the testing personnel, with appropriate skills, to be available during the whole testing lifecycle of the project. During a large project with a long duration, this will require active maintenance of motivation given that same functionalities will need to be retested several times during the project.

A lot of regression testing is needed in every phase to verify that previously developed and tested functionalities are not affected when new functionalities are developed. Retesting of the whole end-to-end process should occur to ensure that the solution works effectively and the user experience is optimal. It is good practice to have common quality reviews and checkpoints in the testing plans for this purpose.

There are various testing methods and tools in the market available to support testing. If the COTS vendor has its own method and tools, it is recommended to use them. Tax Finland used the COTS vendor's testing methodology and testing tool for COTS testing, and the existing methodology and tools for legacy testing. A flexible tool is needed to handle test planning, test scenario assignment, testing progress and reporting as well as error handling and error prioritisation.

### *Testing Automation*

Automated testing is something that should be considered when starting a COTS taxation project. Automated testing is not easy and requires an upfront investment, but can be highly beneficial. Testing automation is its own subclass of robotic automation where a robot software uses another software through its interfaces, including the graphical user interface.

Not all testing can be automated in a COTS project but much can be. Testing a new functionality in the system is, however, seldom something that can be automated. If something does not work correctly when tested, the defect needs to be reported to the developers, fixed and the scenario tested again. In automated testing, the test scenario does not have to be manually executed again. As the number of repetitions and the number and complexity of test scenarios rise, the more beneficial automating the test scenarios can be.

A COTS project usually has several roll-out projects with new tax types and processes going into production in each roll-out. Each time new functionalities go into production, at least some testing is needed for functionalities that were brought into use in previous roll-outs. This regression testing is especially important with large systems, such as a taxation COTS systems, where everything is interconnected.

Regression testing increases the quality of the system and reduces the defect detection time. Regression testing by human testers is very expensive and tedious. A robot is more precise and more efficient in repetitive tasks such as regression testing. A robot is also somewhat quicker in testing, and there can be as many robots testing at the same time.

When the project has ended and everything is in production, even minor development and fixes made to correct defects might need a lot of regression testing. At this point, automated testing will be very valuable.

The necessary upfront investment for test automation can be justified through the production of a business case. The business case can be calculated by estimating the amount of regression test scenarios needed, the percentage of possible automated scenarios and the corresponding amount of manual testers needed. This can then be compared to the costs of the automated testing, including the tools, creation work, execution, maintenance work and reporting. Tax Finland's conclusion was that if the regression tests need to be repeated more often than once a month, then the business case will generally be positive. On top of these savings, there are benefits from increased quality (although this is harder to quantify at the outset). Test automation can also be used for many other things in testing such as test material creation and security testing.

There are several approaches on how to implement test automation. It is important that subject matter experts are responsible for defining the regression test scenarios for automation. Test automation scripts can be written and maintained by third party consultants or the tax administration's own staff. It is good practice for the test automation team to work in short sprints, with reviews (including demos) arranged after every sprint.

The approach Tax Finland chose for test automation was "keyword driven". (In keyword-driven testing, a keyword maps to individual actions to be tested such as opening and closing windows, mouse actions, use of menus etc. The test then simulates these actions.) Tax Finland defined and

implemented keywords such as “Click Text” to be used in writing test automation scenarios. Automated test scenarios can be written without programming/scripting knowledge (once the needed keywords are implemented). Theoretically, it would be possible to reuse the test scenario scripts and just change the programming behind the keywords. However, automating test scenarios with keywords is faster than just scripting. Maintenance of keyword driven test scenarios is easier than maintaining test scenarios automated with programming/scripting.

Tax Finland uses open source tools for test automation and has a growing number of virtual workstations to run the test scripts. Commercial tools are not necessarily needed, since the open source tools are generally very good. Many commercial tools promote test script recording, but recording does not really produce easily maintainable scripts.

Test automation should be started on a small scale, because it takes time and experience for the subject matter experts and other stakeholders to understand the benefits. Concentrating on cost-optimising test automation maintenance, rather than the initial automation work, is a good practice, since that is where the real costs arise. Test data management should also be given priority and is highly dependent on the system to be tested.

#### **Box 4.2. Testing – Statistics for the 4<sup>th</sup> implementation project**

- 74 000 Unique System Test Scenarios Passed
- 322 testers spread across five offices in Finland
- 21 045 conversion verification items checked

*Source:* Finnish Tax Administration (2019)

## **4.6. Usability**

Usability guidelines are a good way to ensure usability and consistency when designing graphical user interfaces (GUIs). The guidelines should be based on common user interface design practices, as well as through usability evaluations and testing.

It is beneficial to have a usability expert working as part of preparation and implementation projects. A usability expert should participate in designing the user interface (UI), including functionalities (e.g. filing processes), the texts displayed in e-services and visual layouts. A usability expert should also plan usability evaluations and conduct test sessions.

Customers should be heavily involved in the development of e-services. The aim of co-designing e-services with the customers is to ensure that e-services are easy to use, self-explanatory and as helpful as possible. Involving customers can result in continuous and quick feedback on open issues and rapid validation of design proposals. In addition to usability tests, regular interviews and walk-throughs with single users and user groups are advisable.

Usability tests are often conducted in individual test sessions, where users representing different user groups, or officers representing different processes, perform predefined test tasks using e-services or the internal system. Such tests are important in verifying important design decisions, process flows, texts, instructions and visual details. Several usability test rounds should be planned for e-services during an implementation project. There should also be usability tests at least for the most widely used internal functionalities.

## 4.7. Training

### *How to Prepare*

Most administrations will have an existing training department or function, and it is recommended that discussions of how to organise COTS related training should take place at an early stage, in particular:

- whether the existing training methods, tools and administrative personnel will be used to provide COTS training
- or whether training should be integrated directly within the COTS transformation project.

Tax Finland chose the latter course. The organisation of training, including content resourcing and scheduling, was an integral part of the implementation project, even though this resulted to some extent in a parallel training organisation within the administration. The existing tools for training sign-up, monitoring and e-learning environments were used, but all of the training-related practicalities were kept within the COTS project. The training department provided support according to agreed co-operation models.

Most administrations will also have their own traditions in training, and it is quite tempting to think that all the existing methods and philosophies should be used also during COTS trainings. However, the vendor most likely has its own methods, tools and techniques, which have been successfully used on other projects. An in-depth discussion is required before planning training approaches to examine how the best practices of both the tax administration and the vendor might be combined.

Finally during the preparation phase of the project, it is important to identify the employees who will be affected by the implementation project, e.g. their roles, number and the location. This will help in planning as well as in targeted communication.

### *Elements of Training*

#### *Training needs assessment and training plan*

Training planning should start with training needs assessment meetings, which should ideally take place during the first months of the implementation project. The purpose of these meetings is to map the new functionalities with the previously identified employee roles. Based on the training needs assessment, the necessary courses, course contents, duration and target groups for each course can be identified. It must be noted, however, that at this stage the content of training will still be on a very high-level, as the development is still going on. During these meetings, the magnitude of the change should also be assessed, because this has an effect on the training method, e.g. is online training sufficient, or will class-room training be needed and what kind of ongoing training will be required.

After the training needs assessment meetings, a training plan should be compiled. This plan should include the high-level schedule, duration and locations for different types of training. In addition, the methods, target groups and course structure should be drawn up, with detailed "training paths" for each role. The training plan should be reviewed by the process owners and approved by the COTS transformation steering group. This way the process owners are both aware and committed to the training process, and senior management can understand and approve what can be quite an intensive and time-consuming training period both before and after the go-live.

### *Trainers*

The number of trainers should be based on the number of employees per local office. Trainers are also a key group in providing desk-side support after training, and the aim should be to have a trainer or two at least in all the bigger offices. It is important to note that training is a resource-consuming exercise. The number of trainers is high and, where they are tax administration employees, they will be away from their operational tasks for a long time. This requires negotiations with the business units, and top management commitment to prioritise the COTS transformation project and to adjust operational tasks where necessary. In order to be able to do so, the business units need the estimated number of trainers as early as possible.

Trainers should have their own on-boarding program, during which they are made aware of their role and tasks. It is of crucial importance for them to learn to use the system themselves. Trainers play an active role: they create the training material and exercises, as well as build the data in the training environment, and of course train their colleagues prior to and after the go-live.

Trainers also have a key role in desk-side support. It is normal, and unavoidable, that there is a lot of confusion right after go-live and the first weeks are not easy for anyone. Trainers have to face the possible frustration of their own colleagues as they get used to the new system, and this makes it important to make the trainers aware of their role in change management in general. It is also to support them in what can be a demanding and stressful task.

### *Trainings*

There are several methods that can be used in training: self-study, online training, class-room training, study groups etc. All methods should include exercises performed in the training environment provided by the vendor. The method should be chosen based on the complexity and/or the criticality of the functionality.

The "perfect" timing for training is rather difficult, especially in a project where development and testing usually continues right up until the cut-over. The system might not be completed when training first begins. This requires expectation management, good communication and the visible commitment of top management.

It has to be accepted that training cannot cover everything and will only be the first step in renewing expertise. A lot of effort and resources need to be put into desk-side support to get as smooth a start as possible. Acquiring new skills does not happen automatically. It is hard work, needs motivation, commitment and resilience from the trainees. It needs to be seen as both an individual and collective responsibility.

#### **Box 4.3. Training – Statistics for the 4<sup>th</sup> implementation project**

- 3 600 end users trained
- 200 trainers trained
- 97 unique courses offered over 560 times

*Source:* Finnish Tax Administration (2019).

## 4.8. Technology

Technology is a surprisingly small topic in a taxation COTS project, but there are still areas that the IT department should be involved in. Naturally, the infrastructure needs to be planned carefully and there needs to be good communication between the COTS provider and the infrastructure provider. An experienced COTS provider should be on top of this.

A key issue is the performance of the database management system (DBMS). With DBMS it is necessary to think of needs at least three years ahead, so it is important to be careful when selecting the DBMS infrastructure. Planning the application server is not so important because it is possible to add more capacity if needed. It is important to take full account of the recommendations of the COTS provider as to the brand of DBMS even though the tax administration might have other preferences. Otherwise, a difficult responsibility discussion might come up if the final performance is not at the right level. Thorough performance tests are critical to find any problems with the infrastructure design.

A taxation COTS project requires several different environments, such as development, testing, conversion, training and production. This applies especially for testing. There are different durations of business processes, different amounts of test material needed and in an integrated system, everything is interconnected. Building and maintaining environments is labour intensive. That is why the supporting features for testing the COTS solution should be given a serious consideration.

The most important feature is the management of separate logical testing materials for different tests and testers within the same physical testing environment. Also the ability to develop and test different versions of the COTS configuration is important, especially in a phased approach, where functionalities that are already in production need to be developed in parallel with the implementation project. Tax Finland's COTS product has a particularly useful solution for isolating test materials between different testers.

Security design is also critical. It might slow things down in the beginning, but if it is not planned carefully, unwelcome surprises might occur later on. It is once again important to plan this together with the COTS provider and the infrastructure provider. There is also a lot of work in getting familiar with the security solutions within the COTS software, which may include new security features.

Finally, as emphasised before, controlling the amount of customisation made to the COTS system is strongly recommended. A clear distinction between 'good site code' and 'bad site code' is needed. Bad site code is code which is not using the features of the COTS software, but replacing them. Good site code means adding code where the COTS provider has originally designed it to be added. Configuration can also be considered as good site code.

Avoiding customization is, at its core, a communication task. Everyone in the project should understand its importance, including the top management. A good method of measuring the level of customisation is needed, which should be automatic if possible. There should also be a decision process in place to ensure well-considered customisation decisions. Tax Finland agreed on a maximum level of customisation (20%) in the contract and measured it automatically.

## 4.9. Cutover and Decommissioning

Cutover and decommissioning (or ramp down) are separate actions, but have common ground in planning and documentation. There are therefore advantages in planning these activities together.

### *Cutover*

Cutover means the period of time when the actual change from using legacy systems to using COTS takes place. This usually happens over several days. Normally it includes the following steps: stopping the systems, conversion (extracts and loads), approving conversion, manual work (e.g. manual updates to systems before going live), “smoke testing” (to make sure the most important functions work) and starting the systems.

Cutover planning should begin with identification of the methods and requirements for cutover (how many cutovers, system outage demands, one or multiple conversions, customer service requirements and other business requirements). Doing this together with all the involved areas and documenting it in detail is highly recommended. Dedicated internal and external communication personnel should also be included in planning.

The detailed plan should include all the steps needed for supporting a successful go-live. Depending on the size of the implementation project, the cutover plan can be divided into multiple sub-plans reflecting the overall organization of the project, e.g. a conversion plan, a technical plan, a legacy plan, a smoke test plan and a communication plan. The important thing is that every plan has an owner, who is responsible for creating the plan and conducting it. Cutover should be practiced in dry-runs and dress rehearsals. Dry-runs are organized walkthroughs of plans, where all the parties involved go through their tasks. Dress rehearsals are actual executions of the cutover tasks. The business people who are in key roles when the conversion is approved and the new implemented functions are smoke tested should participate in the dress rehearsals.

Part of the cutover planning is to create a support plan for the end users. A COTS implementation project always means updated business processes and new tools for the end users, which requires a lot of support for the end users after the go-live. It is important to achieve a smooth start and to create a positive feeling about the new system for the end users to be productive as soon as possible. This requires that the end users get used to the new system quickly and any issues that prevent efficient working are resolved. A support plan should be applied through the whole organisation and should include methods for the detection and correction of errors, communication and the methods of providing end user support.

### *Decommissioning*

Decommissioning is the task of making the legacy systems, all the data they contain, all their documentation and the supporting infrastructure unavailable.

Based on the procurement and the contract it should be clear which legacy systems the COTS system is replacing. If lowering the costs is one of the primary goals of the COTS transformation project, it is likely that most of the legacy systems will be replaced. A high-level schedule of the legacy systems' lifecycles should be drafted as part of the COTS transformation planning. Creating legacy systems decommissioning plans in the same format as cutover plans is highly recommended to prevent misunderstandings arising during the project.

The key thing in decommission planning is to start early enough while all the legacy expertise is still available, including business and IT experts. Once the decision to replace legacy systems with a COTS product is made, there is a high risk that the legacy system experts start looking for new jobs.

Key questions to consider in decommission planning are whether:

- there is a need to store existing data outside of the COTS solution
- the data contains only taxation data or also user log-in data or other data

- legislation restricts the deletion of data, systems or related documentation
- there are contracts that are expiring or technology that will become outdated.

COTS projects tend to take several years and it is necessary to abandon legacy systems as soon as possible. When the legacy systems are decommissioned, the COTS implementation project can focus fully on the new system. On the other hand, risks of the COTS implementation project failing in business critical functions should be considered carefully and the need to keep the legacy systems running for a short period after the go-live should be evaluated.

#### 4.10. Other lessons learned

Finland's tax legislation is very detailed and in some cases very complicated as well. Complicated legislation generally leads to a more complicated implementation, but this is true both in a COTS system and a tailored taxation system. There have not been any parts of the legislation in Finland that have been impossible to implement. It is also worth remembering that there will always be taxation experts that think that their jurisdiction or tax type is the most complicated one that exists, and that it can never be implemented in a COTS system. These experts are a valuable source of information about the most difficult areas, but they might not necessarily be the ideal personnel to work full time in an implementation project.

In a transformation project that consists of several phases and implementation projects, it is important to record the successes and the failures, and evaluate and improve continuously. A good practice is to have lessons-learned workshops on different areas of the project (requirements, testing, training, conversion etc.) after every implementation project. External project auditors can also be used to evaluate and improve the methods, although it is important that the COTS provider understands and sees the benefit of the changes and that they are mutually agreed upon, since they will have in-depth experience from previous projects of what it takes to successfully implement the solution.

## 5. Production Support/Maintenance

### 5.1. Handover from Project to Production Support

Handover is the moment when the project organisation hands over the project results to the production support organisation. After the handover, the project can be closed and project personnel released from their project tasks. At this time, the production support organisation is mature enough to take care of the production. There should be handover criteria in place to determine whether the handover can happen. These criteria should include the following:

- Production support organisation has been nominated
- Production support processes and practices have been defined and trained
- Production support related contracts are in place
- Production support related tools are installed, configured and tested
- Project tasks have been completed
- Use of the new system has started and stabilised.

### 5.2. Production Support Model

The goal of the production support organisation is to ensure that the system works reliably and without errors and serves the needs of the customers and the officers. Any errors need to be fixed promptly and according to priorities. End-users need to be trained and supported sufficiently.

The Production Support Model describes the work and methods of production support and should include at least these things: roles and responsibilities of production support; forums and co-operation processes; a process for handling proposals for corrections and changes; and communication plans.

### 5.3. Roles and Responsibilities

#### *Product Owner*

The product owner is responsible for ensuring that the functionalities of the whole system are developed consistently, in accordance with the tax administrations strategy and goals and that the functionalities serve the needs of the business. The product owner keeps track of the new features of the system and evaluates how to make use of them. He supports the subject matter experts in making best use of the system and in analysing the impacts of the development proposals.

#### *System Manager*

The system manager is responsible for the administration of maintenance work carried out by the developers and the production support process. The system manager directs and administers the preparation of development proposals so that the content, potential workload and technical solutions for all development proposals are available on schedule.

### *Subject Matter Experts*

Subject matter experts (SMEs) are responsible for the content of the functionality in their area, its development and its maintenance. They are responsible for ensuring that the content of the functionality meets both statutory requirements and the tax administration's needs. SMEs are responsible for ensuring that all proposals for developing functionality are processed and that the impacts of a development proposal are analysed and prioritised. They are responsible for the requirements, testing, user instructions and training. Each system area / process should have its own SMEs, the number of which will depend on the size and complexity of the subject area.

### *Super Users*

Super users are expert users of the system. They are familiar with one of the business processes and know how to use the system for the needs of the process. Super users provide user support for the officers; assist users in their daily work; resolve unclear situations; and promote best practices in the system's utilisation. They also monitor the use of the system and user competence and report possible needs for additional guidance or training.

### *Other roles*

Depending on the size of the system, other supporting roles might be needed. These roles can include testing co-ordination, training co-ordination, communication and configuration management.

## **5.4. Production Support Process**

The identification of necessary corrections and suggestions for improvements can come from customer feedback, the tax administration's officers, or from new legislative requirements. If proposals are considered to be worth taking forward, a ticket should be created for each proposal, in order to keep track of the proposals and record all the necessary information. These tickets should contain the title (briefly describing the issue); priority (from system critical to low); type (change, correction etc.); area (system area); and a more detailed description, including an assessment of the impact of the desired correction or change.

The tickets should be prioritised, first in each area and then across the areas. Product owners, system managers, subject matter experts and the vendor should decide on regular basis (weekly, bi-weekly), based on this prioritisation and the available development resources, which tickets should be moved into development. The available resources for production support should be agreed beforehand.

It is likely that there will be more tickets waiting than moving into development, since only the ones that can actually be worked on should be moved into development. Otherwise, the flow of tickets can be disrupted and everything slows down. A method should be in place for managing the workload of production support (for example the “Kanban” method which helps to visualise work flow). Acute bug fixes and disruptions should be processed as urgent. However, the impacts of the fix should always be assessed.

## 6. Last words

Tax Finland's road from tailor-made taxation software applications to one COTS system for all the tax types and taxation functionalities has been a long one, and is still continuing. So far, Tax Finland has had four successful implementation projects and is just starting the fifth, and last, implementation project. Although the biggest benefits will be visible after the last project is finished, all of the taxation tax types are in COTS, and all the legacy systems can be decommissioned. Tax Finland has already seen many improvements and significant reductions in IT costs.

All the implementation projects have had their share of difficulties and moments of despair, but all the problems have been resolved eventually. Tax Finland has not encountered anything in Finland's tax legislation that could not be implemented in the COTS system, nor any major issues in production.

A COTS transformation project is a huge undertaking and it requires continuous commitment and support from the whole organization. In return for this commitment, the organization gets, in addition to the COTS system and its benefits, a number of highly experienced and knowledgeable staff. The amount of knowledge that the people working in the transformation project accumulate is highly valuable and oftentimes such knowledge does not exist anywhere else in the organization. Once such experience and expertise has been accumulated, the administration is able to utilize it in the future as a change force for any task.

# Introducing a Commercial Off-The-Shelf Software Solution

Tax administrations are increasingly looking to adapt and modernise their IT systems to make them more efficient, effective and better able to provide the services that taxpayers increasingly expect. This can be a difficult undertaking as there are often large numbers of highly integrated legacy IT systems carrying out different processes which may not always work seamlessly together. Following detailed market research and after thorough consideration, the Finnish Tax Administration (Tax Finland) decided to replace its many legacy software systems with a single commercial off-the-shelf (COTS) product. The implementation of the COTS solution for Tax Finland in all its phases took approximately ten years. The experiences and the lessons learnt during the implementation process are shared by Tax Finland in this handbook as an aid to others considering similar reforms.