

Overlay Report

Assistance for analysis of “Et fælles inddrivelsessystem”

A large, stylized blue chevron graphic pointing to the right, composed of two parallel lines that meet at a point on the right side.

High performance. Delivered.

Table of content

1	Executive Summary.....	2
1.1	Conclusion & recommendation.....	2
1.1.1	Ambitious Scope and Fully Automated Collection.....	3
1.1.2	Lack of Documentation and Requirements.....	3
1.1.3	Two Complex Integrated Applications.....	4
1.1.4	Pervasive Focus on “the Simple Path”.....	4
1.1.5	Gaps and Functional Limitations.....	4
1.1.6	Lack of Data Governance, Data Validation and Data Quality.....	4
1.1.7	Lack of Consolidated Data Overview.....	5
1.1	Context.....	5
1.2	Analysis Approach.....	6
1.2.1	Accenture’s five analyses.....	6
2	Overlay report.....	8
3	Functional Analysis.....	8
4	Technical Analysis.....	11
4.1	Chronological Context for EFI Programme and Findings.....	11
4.2	Documentation & Design.....	12
4.3	Overall Architecture.....	12
5	PPSM Analysis.....	15
5.1	Organisation and Governance.....	15
5.2	Programme and Project Management.....	15
5.3	Delivery Structure and Processes.....	16
5.4	Supplier Management Structure and Processes.....	16
5.5	Defect and Release Management.....	16
5.6	Incident and Change Management.....	17
5.7	Test Approach and Methodology.....	17
5.8	Transition to Long Term Operations.....	18
6	Data Migration Analysis.....	18
7	What are the Overall Consequences?.....	20
8	Appendix – List of Defined Terms.....	21

1 Executive Summary

From February - September 2015, Accenture has conducted a thorough analysis of "*Et fælles inddrivelsessystem*" (EFI) and the associated system "*Debitor motor inddrivelse*" (DMI) on behalf of *Skatteministeriet*. The analysis includes five in-depth analyses of EFI's and DMI's organisation, functionality, technology, data migration and data.

The conclusions and recommendations are based on analyses of selected areas of EFI and DMI that are key to the collection process. The areas have been selected in cooperation with *Skatteministeriet* and SKAT in order to ensure that they are representative of EFI and DMI as a whole, to the fullest extent possible.

This executive summary and the associated overlay report does not address whether the state of EFI and DMI is in accordance with the contractual conditions in the EFI and DMI contracts.

1.1 Conclusion & recommendation

Based on the results of the conducted analyses, it is Accenture's conclusion that EFI and DMI should be discontinued. Accenture recommends that EFI and DMI are gradually phased out in a controlled process and replaced by a new IT solution. As part of this process, it should be further examined if some parts of EFI and DMI can be used in the interim period, until a new system can be set up. Likewise, the feasibility of establishing a minimal temporary IT solution should also be examined, which can support the areas gradually being shut down in EFI and DMI.

This recommendation is based on the fact that the EFI and DMI have significant deficiencies and errors in most of the analysed areas. Accenture particularly highlights seven key analysis results which are critical in forming the conclusion that EFI and DMI should be discontinued. The analysis results are presented in sections 1.1.1 - 1.1.7

Accenture's conclusion has been made based on a comparison with the general industry standards within system development, similar collection systems in other countries and *Accenture Delivery Methods* (as described in section 1.3).

Accenture's analysis has shown that a significant lack of adequate documentation, combined with a lack of detailed requirements and an inadequate approach to testing of EFI and DMI have resulted in a consistently high level of error. These deficiencies have also contributed to making the subsequent debugging process difficult and time consuming.

The high level of error has been constant since the system went live in September 2013. There have been improvements in some areas, but significant parts of EFI and DMI are still non-functional, and new production errors are identified on a regular basis. EFI had approximately 1000 errors after go-live, and in the subsequent two-year period of operation, it has not been possible to reduce this number of errors significantly. One of the main consequences of this high number of errors is that only very simple cases can be processed using EFI and DMI.

Furthermore, Accenture concludes that any solution to EFI's and DMI's challenges will require a redefined solution with reduced scope and associated detailed requirements, whether the goal is to attempt to make EFI and DMI fully operational or to build a new system.

Based on these analyses, it is Accenture's conclusion that a fix of EFI and DMI would be very risky, time-consuming and costly, hence it is Accenture's conclusion that a fix of EFI and DMI is not recommended. It is Accenture's assessment that a fix would require a drastic restructuring of EFI and DMI and that it would equate to starting over with an entirely new system. The recommendation to discontinue EFI and DMI is based on the seven following analytical results (1.1.1 - 1.1.7).

1.1.1 Ambitious Scope and Fully Automated Collection

The EFI and DMI systems are based on an ambitious vision of creating a fully-automated collections system with minimal manual processing. The scope for EFI and DMI is far greater and considerably more complex than comparable systems in revenue collection authorities in other countries.

The number of different claim types combined with legislation for each claim type makes the collection area a complex area to be supported by the system. For example, the different legislation determines how each claim type should be processed, generates interest and collected in EFI and DMI. This, combined with a large amount of very complex business rules and matrices, makes it very difficult to validate whether the EFI's and DMI's fully automated processes are being executed properly. Proper testing of whether the systems are operating correctly is also difficult due to the very limited documentation available on how EFI's and DMI's processes and functionality should work and how it should be used in day-to-day operation.

Accenture's analysis of the underlying architecture, integrations and services from external systems has identified a number of areas that make EFI and DMI vulnerable. EFI and DMI have a high operational dependency on interfaces and services (due to the service-oriented architecture), which means that when, for example, external systems are out of service or there are other operational problems, this can have a direct impact on EFI's and DMI's availability. Finally, high-quality data from claimants is an important prerequisite to fulfil the vision of fully automated collection. Accenture's analysis has identified challenges with data quality and data integrity in EFI and DMI, which has contributed to EFI and DMI not being able to fulfil the vision of fully automated collection.

1.1.2 Lack of Documentation and Requirements

Accenture's analysis has identified a lack of comprehensive system documentation. As of July 2015, there is no detailed, documented and complete definition of what the System is required to do. Without this definition, it is our opinion that the subsequent phases of activity including design, build and test, which depend on these detailed documented requirements were seriously compromised. Therefore, it is not possible to accurately describe the current state of the System in a normal way.

The purpose of the system documentation is to provide a detailed understanding of how the system (EFI and DMI) is constructed, how it works and how it can be further developed and maintained. Therefore, lack of system documentation makes it very difficult to determine whether the EFI and DMI is working as intended or to design changes to fix issues that are identified. The lack of detailed requirements and end to end process designs resulted in EFI and DMI not being end-to-end tested and the systems were not adequately validated before being put into operation in September 2013.

The basis for managing a testing process is to create tests to verify the documented solution requirements and expectations. If the requirements are not sufficiently detailed and documented, there is no basis for testing the system. The lack of system documentation meant that the testing process has not been optimal, causing extensive retesting of errors and deficiencies that should have been corrected in the first iteration.

1.1.3 Two Complex Integrated Applications

The collection system is divided into two tightly integrated applications, EFI and DMI. Given the lack of *end-to-end* requirements and design, and the lack of documentation of functionality in the two applications, it is very difficult to debug and test how EFI and DMI will function as a total solution. It is therefore very difficult to assess how errors in EFI and DMI can be resolved.

1.1.4 Pervasive Focus on “the Simple Path”

The Simple Path is the most commonly used route through a system where everything works as it should, and there are no data errors. In the analysis, design, development and testing of EFI and DMI, a pervasive approach of focussing on “the simple path” has been employed. Throughout requirements specification, use cases, designs, codes and tests, EFI and DMI are lacking the necessary mechanisms that enable the system to handle the diverse and complex situations that arise in connection with operations outside “the simple path”.

As the functionality of the EFI and DMI does not work as intended and data errors occur the system cannot sufficiently process cases on “the simple path” Only very simple cases can be processed in EFI and DMI which means that many cases are not processed, or must be processed manually.

As the logic to handle cases outside “the simple path” is not in place, the consequence is that EFI and DMI do not sufficiently support the real-world case processing, where data errors, data storage errors, lack of functionality or manual errors can all occur.

1.1.5 Gaps and Functional Limitations

The functional limitations in EFI and DMI affect SKAT's ability to manage debtors effectively, perform the collection process, collect debts before they expire and improve efficiency (such as automation). The lack of functionality also makes it difficult to handle and collect debts in a structured and consistent manner.

It is Accenture's conclusion that the functional limitations in EFI and DMI are significant and this would be technically very challenging, time consuming and costly to fix because of the way the EFI and DMI is designed and constructed.

1.1.6 Lack of Data Governance, Data Validation and Data Quality

Accenture has identified inconsistencies indicating that the rules governing the relationship between data records in relation to the same claim are not present at database level in DMI. There are examples where a claim has no liabilities, which means that the claim cannot be collected. The opposite also occurs where a liability is not linked to a claim. This remains

undetected in EFI and DMI, which means that the collection process is impeded, and that some claims expire or otherwise cannot be collected.

Additionally, there is very limited validation of data coming into EFI. This applies to both the EFI-interface and the web-based interface that the claimants use.

For example, there is limited validation of dates beyond simple validation of the format, which ensures that a date is actually specified. This means that EFI and DMI can receive claims with incorrect information, which therefore cannot be collected. In this way, existing errors in the data from the claimants will be stored and used by EFI and DMI without these being systematically detected and subsequently corrected. Furthermore, after the data has been sent to EFI and DMI, the claimants can edit and modify data, such as the expiry date, amount or other essential information, without their changes being validated. This can cause errors in the subsequent debt collection process.

In connection with the development of DMI, Accenture has observed examples of typical data management methods and tools not being used. For example, this applies to referential integrity (a method of ensuring internal data validity), normalisation (a method to avoid data duplication) and data validation. This leaves the production data unreliable and prone to error.

Furthermore, there are serious challenges concerning data management in EFI and DMI. Accenture was unable to ascertain from the analysis whether an assessment of data quality was prior to or in connection with the migration of data. It is Accenture's conclusion that the process and methods that have been used in connection with the migration of data, has led to significant errors in SKAT's collection.

1.1.7 Lack of Consolidated Data Overview

Data is distributed between EFI and DMI. Accenture has identified that mechanisms that are in place to ensure reliable transfer are not used consistently. Mechanisms that are often used in similar systems to detect inconsistencies are not in place. This means, for example, that it can be seen when new claims are received in EFI and transferred to DMI for subsequent processing. An error in the transfer between EFI and DMI can result in a difference between what the claimant believes is being collected and what is actually being collected. This is also evident when the treatments are managed in EFI and claims are managed in DMI, without there being any mechanism to ensure that information about the treatments and debts are always synchronised.

The key information about expiry dates for claims is registered in at least two places in DMI. These dates are not always identical, which is critical in determining whether a claim has expired. This can lead to errors in assessing a claim's enforceability.

1.1 Context

In 2005, it was decided to centralise the collection of 490 types of public debts under one body, SKAT. Between 2005 - 2013, the collection was supported by a number of different software systems. A new single unified debt collection system, comprised of two main applications - EFI and DMI – was initiated in 2013 with the objective of providing an improved and highly automated collection service.

EFI and DMI have never functioned as intended, which has resulted in a significant reduction of SKAT's ability to collect debts and avoid expiry of debts.

1.2 Analysis Approach

Accenture has conducted five analyses of the organisation, functionality, technology, data migration and data in order to establish a reasonable foundation for the conclusions and recommendations.

The results of the analyses have been compared with collections systems in other countries as well as general industry standards within system development represented by *Accenture Delivery Methods*, which is Accenture's globally used methodology for organising, planning, design, implementation and operation of complex IT projects.

Accenture's analyses are based on data collected through interviews and workshops with key employees of SKAT and external suppliers, as well as review of available documentation, including process and technical documentation and source code. In addition to this, Accenture has also conducted a review of selected functional areas and conducted quantitative analyses of data in EFI and DMI. The latter was performed in cooperation with Kammeradvokaten, where it was relevant.

1.2.1 Accenture's five analyses

The analysis of the EFI Programme, Project and Service Management Organisation has evaluated the completeness and quality of SKAT's governance, processes and procedures in the development and operation organisations. The analysis has compared SKAT's governance, processes and procedures in both the development and operations organisation to the *Accenture Delivery Methods for Programme, Project & Service Management*.

As a consequence of EFI and DMI's scope and complexity held up against the planned timeline for the analysis, the functional analysis used a sample analysis approach where selected functional areas in EFI and DMI have been analysed in detail, while remaining functional areas have been assessed more generally. The functional areas were selected in cooperation with SKAT in order to ensure that they were representative of an overall assessment of EFI and DMI. The functional analysis identified selected key processes and examined the extent to which they are supported by the functionality of EFI and DMI. In addition, the analysis assessed the quality of functionality compared to collection authorities in other countries.

The technical analysis aimed at identifying the current state of EFI and DMI and to explain why EFI and DMI has ended up in this state. The technical analysis has assessed the architecture, code and technical documentation, as well as identified the software development lifecycle from initial requirements specification through design, development and testing to finally going live.

The data migration analysis mapped the data migration process prior to EFI and DMI being put into production. The process was then benchmarked against the *Accenture Delivery Methods for Data Migration* in order to assess the completeness and quality of the data migration.

The data analysis, which was conducted in cooperation with Kammeradvokaten, reviewed and categorised more than 30 million claims in EFI and DMI. Among other things, the work

has included analysis of expiry related issues, analysis of the claims' actual collectability, as well as a validation of whether the individual claims are enforceable. The data analysis also focused on assessing the completeness and quality of data in EFI and DMI.

2 Overlay report

The following overlay report is based on the most important findings and conclusions from each of the five analyses that Accenture has performed. The conclusions from each of the five analyses should not be read independently, and Accenture's overall conclusions and recommendation is based on an overall assessment of the conclusions in each of the analyses. Key stakeholders have continuously contributed with inputs and validation throughout the analysis process, however, SKAT has had limited opportunity to validate Accenture's final reports.

3 Functional Analysis

The purpose of the functional analysis was to provide an overview of the functional areas that are supported within the EFI and DMI System and to then assess selected critical areas of the EFI System to understand whether there are functional issues within these areas.

The functional analysis found that critical areas of a standard collection system are missing (i.e. management at debtor level, case and work management, auditing, asset repossession, insolvency & tagging of debtors), and that specific functional areas have serious shortcomings (calculation of expiration dates, reconciliation to ensure data quality, accounting consistency and integrity between systems)

A further complication from a functional standpoint is that the EFI+DMI System was designed to run an end-to-end automated collection process. This means it is complex and that it is not always possible to fix isolated areas of the System.

Following is the functional area overview that was created as part of our assessment work. It outlines the main functional areas used at SKAT in their debt collection and management processes. This diagram below has been updated to highlight the areas where we have found the largest gaps in terms of functional capabilities. The red dots indicates that the process has gaps and requires work, green indicates a high priority core component and light-blue a high priority supporting component.

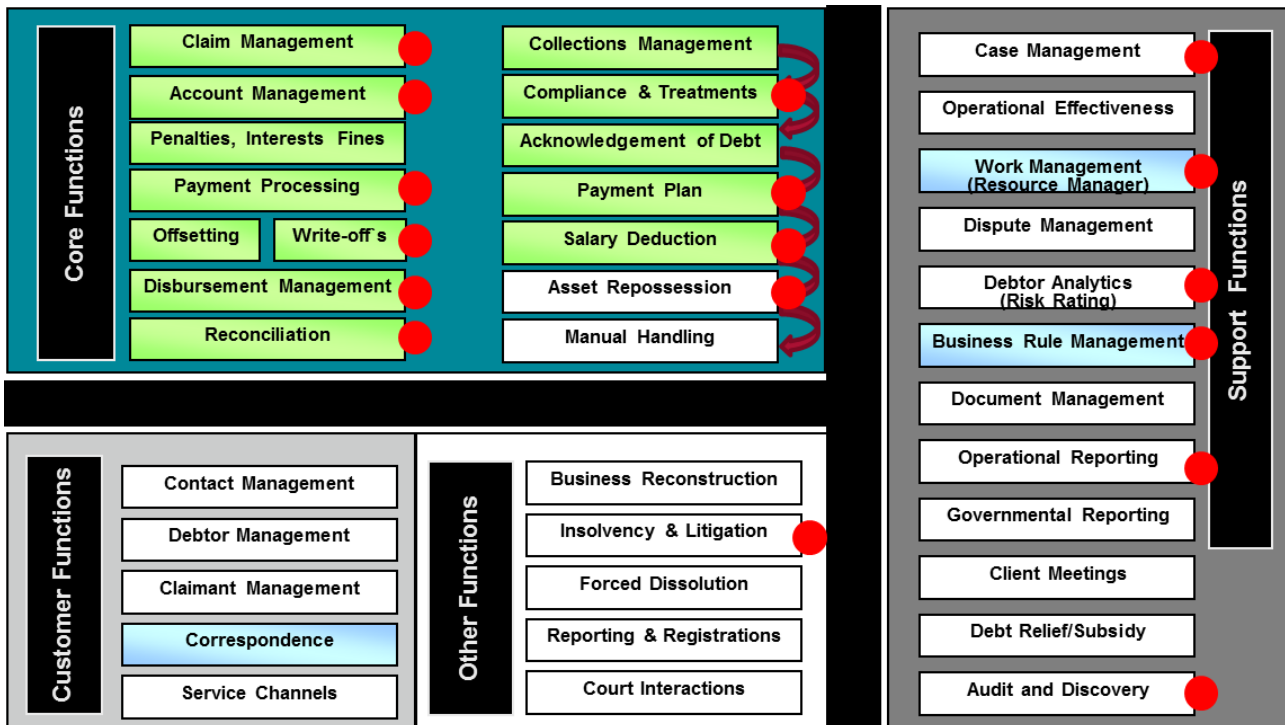


Figure 1 Functional Area Overview Highlighting Large Functional Gaps

In more detail our findings show that there are gaps and functional limitations, which are time-consuming, risky and technically challenging to fix. These gaps and limitations are impacting SKAT's ability to manage debtors effectively, perform their collection processes, collect debts before they expire and improve efficiency (e.g., through automation). The functionality also makes it difficult to manage debtors and cases in an integrated uniform manner.

Following is a summary of our key findings:

- There are a large number of claim types in the System that increases the complexity of the System and the treatments.
- The System is not build to handle expired claims that has not been written-off, which has the effect that the System will apply treatments and credits to expired claims.

There are significant issues with the management of claims including the handling of claims on treatments and with the rules that govern claims. Additionally, the 490 claim types lack grouping and categorisation, which is important for consistent handling and re-use.

- The functionality is complex and there is very little re-usability in areas such as treatments
- The quality of the data is compromised by the functionality.
- It is difficult to implement most changes including critical ones that are urgently required due to the functional architecture and system functional distribution.
- Limits with the functional architecture and design make it difficult to fix functionality in EFI such as adding new claims to treatments, applying debtor level actions (including

tagging), handling of claims and sub-claims, performing updates to treatments and complete transaction roll-back.

- EFI does not support the following business process requirements; case and work management, auditing, mail, asset repossession, insolvency and tagging of debtors.
- There is insufficient reconciliation to prevent data quality issues, ensure accounting consistency and maintain integrity between systems.
- History and logging is insufficient to track actions already taken.
- Auditing is challenging and the functionality does not support comprehensive traceability.
- There is little to no validation on data such as claims and sub-claims, missing business rules and issues with the management and handling of claims including the calculation of expiration dates.
- There is currently no understanding of a fraud detection mechanism in the System.

4 Technical Analysis

The Technical Analysis is based on review of selected parts of the EFI and DMI Applications, as well as selected processes undertaken during the development of EFI and DMI. The report is based on our experience and assumption that the conclusions in the report are representative also for the part of the EFI and DMI Applications that have not been reviewed.

Our report does not give a full picture with regards to the current state of EFI and DMI, the reasons that have led to the current state of the Applications (EFI and DMI), or if the current state of the Applications are consistent with the original contractual requirements as described in the EFI and DMI contracts.

The scope included the analysis, design, build and test of the EFI and DMI Applications, and the combined EFI+DMI System. Most of the analysis was performed on a sample basis.

In the following sections we have listed some of our key findings and conclusions.

4.1 Chronological Context for EFI Programme and Findings

Was the original planning around the programme execution feasible from a technical perspective? In order to help answer this the events analysed in this report are depicted on the timeline below. Events depicted with grey are unconfirmed dates, where the EFI Programme has not confirmed the exact dates.

There are a number of key points with regard to the overall timeline:

- EFI and DMI were separate projects, performed by separate vendors, and managed by SKAT.
- The most significant is that the design of EFI and DMI was not performed in a tightly coordinated manner. Given the number and the complexity of the dependencies between EFI and DMI, close coordination would have been required between the Applications for the System to integrate correctly
- Overall, there was poor stage containment (completing and closing phases of work, before progressing to the next phase) both within the EFI and DMI Applications, and overall for the combined EFI+DMI System

In our opinion, the original project timeline was not feasible due to these technical dependencies. The iterative nature of resolving the dependencies resulted in additional work and suboptimal design.

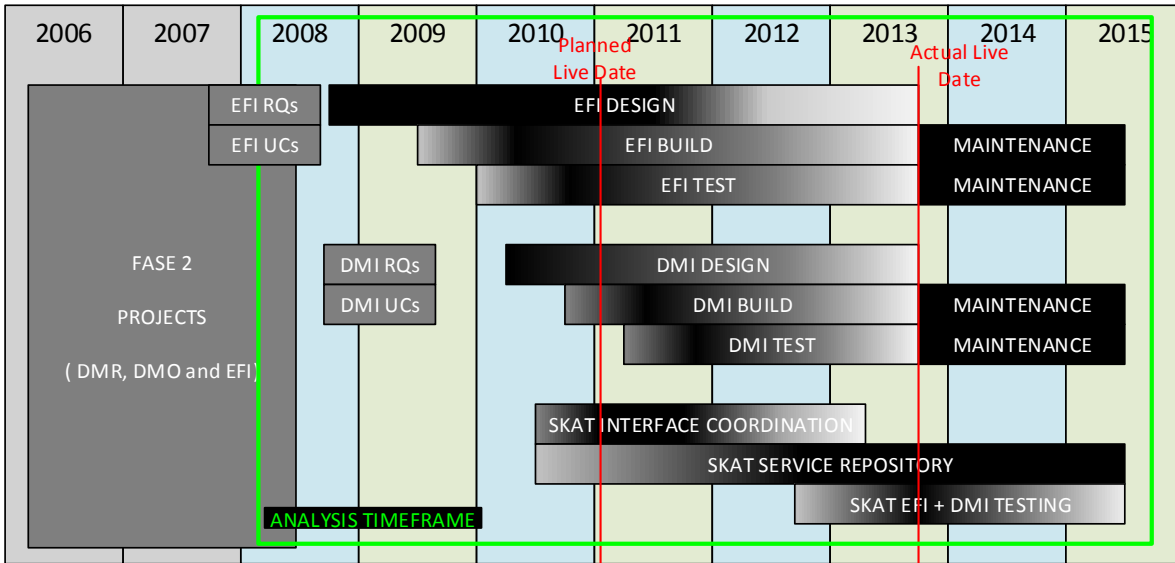


Figure 2 EFI+DMI Timeline

The below sections outlines some of the key findings from our technical analysis.

4.2 Documentation & Design

- Because of the lack of detailed documented requirements, it is not possible to accurately describe the state of the System
 - There is no set of detailed, documented requirements for the EFI or DMI Applications or the combined EFI+DMI System
 - Normal practice is to describe the completion state of a system in terms of the percentage of detailed documented requirements that it meets, validated by testing
 - Therefore it is not possible to accurately describe the current state of the System in a normal way, or to ever complete the System
- During the design of EFI and DMI there was no documented start-to-finish process description of how the overall solution should work to process claims. The high level designs that did exist (use cases) were separate for EFI and DMI. They were not integrated, and they were also not used to validate that the end-to-end solution worked

4.3 Overall Architecture

EFI and DMI are separate applications. They are developed and deployed as separate technical components. Together EFI and DMI are a single system with integration into the SKAT IT estate. Some of the key findings and observations are:

- Neither EFI nor DMI can perform any useful collections function without the other. For example, screens that display the details of debts or interests require EFI and DMI to interact
- It is our opinion they should therefore be considered as a single software system that should have requirements, processes, designs, and tests for the System
- When EFI or DMI require replacement, there will be either a very large body of work required to re-implement the same interfaces again on the new component, or both EFI and DMI must be replaced together

The debt collection system is separated into two tightly integrated applications. In our experience collections systems are usually implemented as a single application. As a result of this separation, performance, reliability, maintenance costs and functionality are negatively impacted.

EFI and DMI has been designed with the aspiration of providing future flexibility. Some flexibility has been provided, however it is not clear this level of flexibility was required. Most of the flexibility has not in fact been used to date. It is not clear all the functionality controlled by the flexibility actually works.

The interfaces between EFI and DMI and the systems they interact with including the ones within the rest of the SKAT were insufficiently designed. This led to a number of problems such as:

- Extensive, ongoing change being required over a long period to fully evolve the interfaces
- EFI and DMI Applications taking more time and effort to finalise
- Making the testing of realistic paths through the code difficult due to being unable to create accurate program stubs. Stubs are normally used in large system implementations to test components or applications well before end-to-end (complete system) testing is performed. Realistic stubs, that enable testing of more than the simple path depend on detailed designs

EFI does not implement rigorous external validation on input data, from external systems or end users. EFI+DMI is partially or completely unable to process records that are missing key fields, and lacks mechanisms to manage the consequences of this. As a result, the System cannot process some real world data.

Most automation in the System has been disabled since the go-live. The event based processing approach, used for "monitoring" results in excessive numbers of events and changes from a customer perspective. In practice, this means that if the automation were enabled as originally designed and implemented, customers would receive excessive numbers of communications and events from EFI+DMI.

DMI is the financial engine at the core of the System. Unusually for a financial application it does not enforce referential integrity. This has allowed DMI to store invalid records. This reduces the data quality in DMI and the overall integrity of financial processing in SKAT collections.

DMI represents an unusual approach to an SAP Revenue implementation. There is a low usage of SAP package functionality combined with an unusually high percentage of custom objects. The consequences of this are that the licensed SAP product functionality is relatively lightly used. DMI is in fact, mostly a custom application, rather than an SAP application. This will impact maintainability over the lifespan of DMI.

The source code for EFI is of slightly better than average quality; the source code to DMI slightly lower than average quality. The quality refers to technical aspects of the code such as complexity, error handling, maintainability, cohesion and commenting. The source code quality is not a primary cause of problems with the System.

Given these findings, we have concluded that:

- The overall testing approach was compromised both by the lack of detailed documented requirements to test for, and the test approach. Analysis of the testing records confirmed with the EFI Programme test manager shows that 53.8% of the originally planned tests were executed. 52.3% of the original tests (80.5% of non-descoped tests at go-live) were recorded as passing testing. This resulted in a substantially untested System being put into production use
- The scope of the System was not well defined. The complexity of the Original Needs was not well described in the Original Requirements and use cases, or at any subsequent phase. It is our opinion that the complexity of implementing the variety of debts and legal persons, combined with the extent of automation was never fully understood. This resulted in essential functionality being omitted from the System
- There was a pervasive approach through the analysis, design, build and test of the System to focus on the “simple path”, the most common path through the System. Throughout the Original Requirements, use cases, designs, code and tests the System is missing the detail required to process the variety and complexity of situations encountered in production. This also resulted in essential functionality being omitted from the System
- There was no design (solution blueprint) for the overall EFI+DMI System, either at the start or since. Although some materials exist these fall far short of what would be expected to design a system of this complexity. Based on this lack of documentation it is our opinion that the system level “SDLC” activities were largely not performed. This resulted in problems integrating EFI and DMI, causing delays and defects to the overall EFI Programme. Error logs state that EFI had approximately 1000 errors after go-live
- There was an assumption that the System would receive valid data from the internal and external systems feeding EFI/DMI with claims, and also from all users inputting data manually. This assumption is identified because (a) the System does not do normal levels of validation and (b) it was stated in requirements workshops that the EFI Programme had no mandate to control incoming information. This does not appear to be a reasonable assumption, based on our experience of other systems, and the fact that non-valid data has been input to the System. This resulted in the System failing to process correctly when presented with incorrect production data

5 PPSM Analysis

The purpose of the Programme, Project and Service Management (PPSM) analysis was to conduct an analysis of the EFI Programme organisation in order to, in a relatively short time frame, propose improvements to the programme structure and governance for development, service transition and IT operations.

The PPSM analysis concludes with numerous shortcomings in the organisation and tools compared to a standard IT organisation. Our analysis has identified twenty two areas of possible improvement and below we have listed 8 of the key areas where Accenture is actively taking action to improve the Programme, Project and Service Management processes.

5.1 Organisation and Governance

Based on input throughout the interviews with EFI Programme team representatives our conclusion is that the roles and responsibilities within the EFI Programme are not formalized, documented and communicated.

Lack of clear leadership and direction of the organisation affects the overall ability to deliver the EFI Programme to completion. The informal team structure and the lack of clear governance model leads to uncertainty regarding responsibilities and accountability. In the current state, our observation is that, meetings are attended by representatives across multiple teams to ensure relevant information is captured and appropriate actions taken. We have not been able to identify documentation stating business priorities and decisions made by business representatives; hence, there is limited possibility to verify that the EFI Programme is delivering according to business needs and prioritization.

5.2 Programme and Project Management

The guidelines for project managers are documented in "EFI Governance". This document describes project governance (within the EFI Programme and towards the steering committee), and the three mandatory documents Project Initiation Document (PID), weekly status report and closure report. Based on our experience we would expect a more rigid framework around the projects covering each phase of the development life cycle (Analyse, Design, Build, Test and Deploy) especially given the level of complexity of the solution and that the system is already used in production.

Based on interviews with the programme management as well as PMO, our conclusion is that there is no official process for managing and approving changes in scope, timeline or estimate of projects. The Project Initiation Document (PID) is continuously updated, without traceability to the original version, hence it is impossible to determine what changes have been made during the lifecycle of the project. The PIDs uploaded and available on the EFI Programme SharePoint have been reviewed, confirming this conclusion.

The main programme plan, *Hovedtidsplan*, is updated monthly by the PMO, but since the original timeline and scope is not used as a baseline, the programme plan only present the current status without reference to changes within the EFI Programme. The plan does not provide insight to priorities or how resources are allocated across the EFI Programme.

Risks and issues are to be reported for each project according to the guidelines in “EFI Governance”, stating that risks and issues are to be raised in the weekly status reports. Based on interviews with project managers in the EFI Programme, our understanding is that, there is a lack of transparency how and if these are escalated to the steering committee and mitigating actions agreed and executed.

5.3 Delivery Structure and Processes

Throughout the analysis, we have only been able to identify a limited number of IT processes and procedures defined and documented. Our understanding is that the EFI Programme does not have end-to-end control over delivery due to the commercial situation with the suppliers. This has a direct impact on programme delivery; unpredictable scope and timelines and inefficient resource management. For example, the EFI Programme is only informed of the final scope of the weekly release on the day of deployment. Our observation, based on interviews with the PMO and project managers, is also that project timelines are continuously revised and postponed, which also supports this conclusion.

Not having a development framework in place covering all aspects of the lifecycle (from Analyse to Deployment) might have an impact on the quality of releases. If components are built without a documented, detailed design (validated and approved by the EFI Programme), there is a risk that issues are identified and new faults introduced in a later stage than it would have been otherwise (both test and production).

The impact of not having well defined and documented IT processes with clear roles and responsibilities have resulted in overly complex processes and procedures with multiple teams involved at different levels, as described throughout our interviews with the EFI Programme. The absence of a change management procedure, for example, has a direct impact on the control of the production environment, allowing changes to be performed without formal approval and communication to affected parties.

Given the complexity of the solution, in our experience, we would expect significantly longer release cycles with clearly defined test cases and acceptance criteria defined prior to the testing by the EFI Programme begins. As a consequence new functionality deployed has not been regression tested end-to-end to avoid having issues and new errors introduced to the production environment.

5.4 Supplier Management Structure and Processes

As a result of the state of the solution (EFI and DMI Applications), the commercial situation with the suppliers is complex (particularly with *Konsortiet*). We have not been able to find evidence that indicate that there is a common understanding of defined acceptance criteria for deliverables and handover points.

Our overall conclusion is that many activities are informal and reactive in nature rather than proactive, standardized and controlled in advance.

5.5 Defect and Release Management

There is an overview of defect and release management process documented in “Defects, defect management and release management”, however the description is kept at a high-level without clearly defined tasks and assigned roles and responsibilities. Weekly releases

are deployed into production after a go/no-go decision made by the EFI Programme team. Throughout the analysis, we have not been able to identify a formal set of criteria to make the go/no-go decision. During interviews with the Test team it has been described that due to the way the code is structured in development, a go/no-go decision covers the entire release, it is not possible to separate different components in the release and only deploy selected parts.

Based on information gathered during interviews related to Release, Defect and Incident Management as well as Testing; defects and Incidents are prioritised by EFI Programme (in ITSM and QC), but the suppliers have ultimate control of what is finally included in the release as a result of the commercial situation. As described in the document "Defects, defect management and release management", the scope of the release is communicated by the supplier a couple of days prior to the deployment to the integrated test environment, however a new release note is communicated the day of deployment presenting the final scope (these tend to differ some extent).

As far as we have been able to identify, there are no metrics collected on how the releases helps stabilize the solution, such as increased productivity in production.

5.6 Incident and Change Management

Throughout the analysis we have not been able to identify documented processes or procedures for most of the core IT processes, for example Incident and Change Management. Based on this, our conclusion is that documentation does not exist.

The *Supportkæde* (high-level incident flow) is documented indicating a complex support structure with multiple layers of involvement from different stakeholders. A high-level process is described in the *Fælles Samarbejdshåndbog* (supplier cooperation manuals) but our understanding from interviews with Supplier Management, this has not been approved and implemented.

Based on information captured in our interviews within Release and Incident Management, our understanding is that most changes that are introduced into production environment come through the weekly releases. There is no separate process to manage emergency changes into production (for example to resolve high priority incidents). Based on our interviews with the Programme team, standard changes such as changes to matrices and tables, as well as data cleansing activities (that is not included in a release) are not documented, approved and communicated in advance.

5.7 Test Approach and Methodology

We have not been able to identify a formal structure for test execution. There is a Testing Strategy, which has not been updated since 2012. This still has some relevance according to the EFI Programme Test team, however this has not been confirmed throughout the analysis phase.

There are test cases documented in QC. Based on interviews with the Test team, our understanding is that these are used to a varying degree when performing tests of the weekly releases. Testing is performed both by the suppliers and within the EFI Programme following the weekly release cycle, as described in the "Defects, defect management and release management" document, the Test team within the programme has 2-4 days to test each release. We have not been able to identify formal acceptance criteria defined for testing

activities are handed over from the supplier to the EFI Programme for testing of the weekly release. Test environments are not sufficiently integrated, hence it is not possible to execute end-to-end tests before deployment to production. As a result, the “1-2-5” testing is required to validate the functionality in the production environment.

Based on our interviews with the Test team, our conclusion is that there are no documented procedures in place for housekeeping activities related to environment and refresh of test data.

5.8 Transition to Long Term Operations

There is an ongoing initiative to transition the solution to IT Operations, however based on the information we have received from interviews with the project managers of the initiative; all involved parties (SKAT IT Operations, Business Intelligence, Process Owners) have declined to take ownership of the solution in its current state due to missing functionality and missing documentation on processes, interfaces, roles and responsibilities.

High-level acceptance criteria have been defined (described in the document “*Overdragelse aktiviteter og dokumenter*”), but not agreed and approved by the receiving organisations.

A re-organisation has been undertaken by SKAT and the embedding of the processes and tools to address the above shortcoming is underway. This will need to form the platform for a proper way of working.

6 Data Migration Analysis

The data migration effort to convert the data from the legacy systems to EFI+DMI has been analysed at a high level. It is a complex, and not well documented process, but it is clear that it falls significantly short of normal practice.

The most significant root cause of the problems with data migration is that the migrated data was not tested and proven to work end-to-end successfully with the new EFI+DMI System. Critical exit criteria of the migration trial PK4, as stated in the Conversion Plan, were not satisfied. The PK4 test completion report still concluded with a recommendation that PK4 be approved and that the project could continue to the production migration, with prioritized risk reduction actions and defect corrections.

In proceeding to go-live, the EFI Programme went from a large IT project issue to a large business issue. The following diagram summarises some of the shortcomings:

Phase	Area	Finding
Analyse	A1: Identifying Detailed Requirements	Insufficient detail to define scope
	A2: Scope all data source rules	
	A3: Define clean-up	
	A4: Define scope & approach	Strategy documented but not updated. Conversion plan defined entry/exit criteria but was not reflected properly in the migration trial plan. Implications of changes to strategy was that a full analysis was not completed.

Design	B1: Design migration (maps)	Insufficient detail in the mapping documents
	B2: Design tests for migration	
	B3: Non-functional design & tests	
Build	C1: Build & Unit test for migration programs	
Test	D1: Test new system with prepared data	New system was being modified during all migration trials hence the validation/testing done during trials were not reliable as final validation.
	D2: Repeat trial migration with 100 % source data The System was not end-to-end tested on migrated data No evidence was found of clean-up of data claimants in the source systems prior to the data migration trials.	Significant issues identified in test runs not resolved before go-live. Status of trial migrations reported as green despite many shortcomings that were to be fixed or verified. Final trial migration (PK4) identified issues that should have stopped the go-live including: <ul style="list-style-type: none"> Executed on 49 % of data, not 100 % Major functionality not usable (AKR, BEBB, automated treatment assignment) Assessment was that issues could be resolved post go-live. Cleansing effort by <i>Fordringshavere</i> reported as green, but with many remaining tasks listed.
	D3: Test new system with converted data	The System was not end-to-end tested on migrated data No evidence was found of clean-up of data claimants in the source systems prior to the data migration trials.
Deploy	E1: Go-live decision <ul style="list-style-type: none"> Mgr test 100 % data successful End-to-end test of new System converted data successful 	Normal preconditions not met. A recommendation to go-live was provided to steering committee on basis of the PK4 results.
	E2: Final migration run	New issue arose in production migration that should have been found and resolved earlier.

Table 1 Key Areas of Data Migration Issues

7 What are the Overall Consequences?

In light of our analysis and key findings, our conclusion is as follows:

- Based on the functional, technical, data and legal analysis there are serious issues throughout the EFI+DMI System. If there is further analysis, it is probable that further issues will be identified
- It is impractical to resolve the issues with the EFI+DMI System without a rebuild, as it is necessary to define requirements in detail, before revisiting design, build and test. This is therefore a lengthy and costly exercise.
- EFI and DMI should therefore gradually be phased out in a controlled process and replaced by a new IT solution. As part of this process, it should be further examined if some parts of EFI and DMI can be used in the interim period, until a new system can be set up.
- The feasibility of establishing a minimal temporary IT solution, which can support the areas gradually being shut down in EFI and DMI, should also be examined

Areas to Address at a High Level

- Start the process to develop a new IT system to support collection within a new IT organisation
- Fix the data, both preparing white claims for collection and continuing retracing the grey claims to either data cleanse them to become white or confirm them as black for write-off
- Put in place the operational processes for:
 - Repayment of wrongly collected debt
 - Write-off of black claims
- Establish a new operational process for collection with limited support from EFI+DMI, needing a more effective Model Office and a new fit for purpose set of minimal IT components – applications supporting a manual collection process and ad hoc tasks.

8 Appendix – List of Defined Terms

Term	Definition
Aging	When a claim exceeds its expiration date
AKR	Unknown customer number.- assigned to customers that do not have a CPR or SE registration
Application	<p>An application is an executable software program that performs a function or group of functions. It is typically composed of a single technology, and may be integrated with other applications.</p> <p>The term “application” is used to describe EFI and DMI. EFI and DMI are technically separate and are built on different technologies in separate projects.</p>
Automation	<p>Automation is the practice of using applications and other IT to perform tasks that would otherwise be performed manually, i.e. by people.</p> <p>A typical example is the issue of reminder letters to debtors.</p>
Business Process	A Business Process (or Business Process Model) is a formal description of how a business function is performed from beginning to end. The description should indicate the activities performed by users and applications, and the interfaces between each.
Claim	The term used to describe an outstanding debt that has been received by the EFI system
Claimant	The source agency who submitted a debt to the EFI system
Code / Source Code	The “code” or “source code” refers to the human readable instructions that are either compiled and/or interpreted to form object code that can be executed by a computer system.
Collections	The actions related to recovering a debt
CRM	Customer Relationship Management.
Data Cleansing	The actions related to fixing incorrect data
Data Quality	The degree of correctness of the data
Database	A database is a platform application that enables data to be stored and retrieved. Additionally, a database can be used to guarantee the integrity of the data stored within certain constraints, using ACID transactions and Relational Integrity.
Design	Design is the process of converting requirements or a higher level design into a more detailed design. Designs are typically decomposed over multiple levels from a Solution Blueprint through high level design to low level design.
DMI	“Debitormotor Inddrivelse”
EFI	“Et Fælles Inddrivelsessystem”
EFI Programme	<p>The programme of work (a number of related projects) to deliver a new debt collection system for SKAT.</p> <p>The EFI Programme included the EFI project, DMI project and a number of other smaller projects or packages of work to integrate EFI and DMI with other systems within SKAT.</p>
End to end	<p>End to end (e2e) refers to a complete process and/or the supporting IT system. For example:</p> <ul style="list-style-type: none"> The end to end process of handling a Claim includes initial receipt, performing collections treatment(s) and finally closing the Claim.

	The corresponding IT system(s) to enable this process includes all the IT systems and applications that integrate to perform the overall function.
Expiration date	The date on which the claim can no longer be collected
Flexibility	Flexibility refers to the programme principle to implement a system that enables future business requirements to be accommodated primarily by configuration of the System, rather than source code changes.
Simple path	In a software system, the simple path is the default path through the System, with no exceptions. E.g. case processing of a single, simplistic claim with low complexity and no errors along the way.
Interface	An interface is the point where an application connects to something external to the application. Interfaces may be implemented with a wide variety of technologies including files, database tables, web services and more.
IT	Information Technology
Maintainability	Maintainability is a non-functional characteristic of a system. It refers to the ability to make changes to an application over its lifetime to accommodate changing requirements.
Need	As used in this document, the Needs are the detailed documented requirements established through requirements gathering workshops for the selected sample areas. <ul style="list-style-type: none"> The Needs were gathered on the basis that they described what the users originally expected or required the System to do.
Original Requirement	The Original Requirements are the requirements that were used, together with the Use Cases, as the start of the design, build and test processes for EFI and DMI. The Original Requirements we have reviewed as part of this analysis are documented in Section Error! Reference source not found.
Referential Integrity (RI)	Referential Integrity (RI) is the ability to enforce basic business rules at the database level. RI helps ensure the integrity of data by guaranteeing that fundamental constraints on data (i.e. business records) are met. As an example, RI can enforce that every Claim has at least one Liability, or that every Debtor has an Address.
Requirement	A requirement is a formal statement of what a system must do, in order to be working correctly. Typically, the scope of a business application is defined by a number of requirements. Large systems often comprise 3,000 – 10,000 requirements. An example could be “The decision letter to debtor must in all cases contain size, period, type, and due date of all claims covered by the decision.”
SAP	SAP is package software for performing many common business functions.
SDLC	Software Development Life Cycle. SDLC is used to describe the process of creating a software system.
Service	A Service (or web service) is a self-contained unit of functionality and data that performs some useful function. An example Service could be a service that allows users or other applications to check the registration number (CVR, CPR or AKR) for a customer.
SOA	Service Oriented Architecture (SOA) is the concept of creating systems based on integrating Services that provide self-contained functionality.
Steering Committee	A committee responsible for programme management

System	<p>A System is one or more applications that perform an overall business function.</p> <p>An example system is an email system that performs all receiving, storing and sending email (although this may be comprised of a number of discrete applications).</p> <p>The term “system” is used to describe the integrated combination of EFI and DMI, which provides the overall debt collection IT function for SKAT.</p>
Test	<p>A Test is a documented procedure that can be performed to validate compliance with a requirement.</p> <p>As an example, with reference to the definition of Requirement above, the corresponding test would validate that in all cases the decision letter contained the necessary details and that these were correct.</p>
The System	EFI+DMI System
Use Case	<p>A Use Case is a description of the steps that a number of users (actors) must perform in order to complete a business scenario.</p> <p>Use Case descriptions can be used, together with other designs, to provide a design for a system or application.</p> <p>Use Cases typically describe the “simple path” and any number of alternate paths through the system.</p> <p>Separate sets of use cases were used to describe the functionality for the EFI and DMI applications.</p>

Table 2 List of Defined Terms