West Trax

A Cost / Benefit Analysis

Of

SAP Systems

2006

STUDY PRODUCED

BY

WEST TRAX AND EXPERTON GROUP

01/2006

CONTENTS

COPYRIGHT

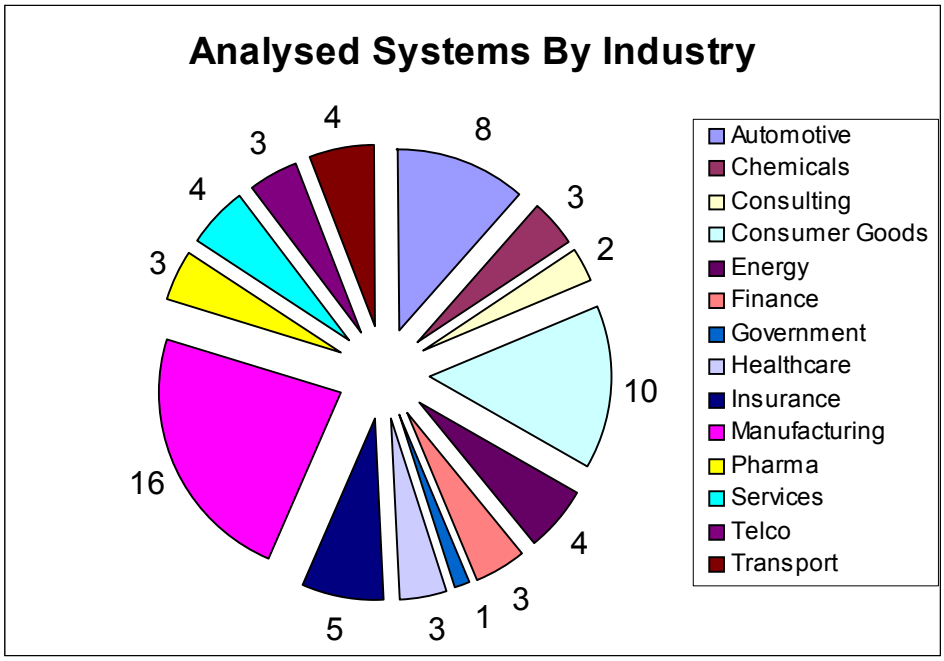
This analysis was performed by West Trax in co-operation with the Experton Group AG. Whilst all data and information has been collected conscientiously and with due diligence completeness and accuracy of the analysis cannot be guaranteed.

CURRENT SITUATION AND ANALYSIS OBJECTIVES

Currently a key priority of many companies is to reduce IT costs and increase the value that systems contribute to the business. The ongoing evaluation, optimisation and modification of IT systems can have a critical impact on the success of a company. For many organisations SAP systems are a major component of the IT budget. They are considered to be a critical success factor in the achievement of business targets.

West Trax produced this study to identify potential cost savings and system productivity improvements for organisations using SAP. The analysis is based on a body of benchmark data developed by West Trax, using the **KPI SCAN®** methodology which was developed specifically for this purpose. It provides SAP customers with strategic and operational information that helps them use their systems more effectively and improves their alignment with the business.

The study analyses data from 93 SAP systems, operated by 69 companies, in 14 different industry sectors. The intent was to objectively examine the actual usage of SAP systems by quantifying Key Performance Indicators in the areas of Costs, Productivity, Performance and Quality. The results identify the degree of standardisation in the analysed systems and the potential for optimisation. Comparisons are provided within and between industry sectors.

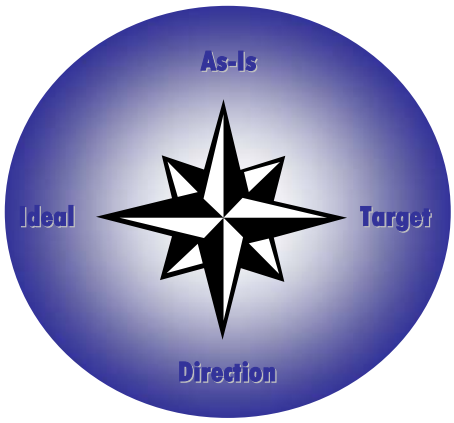


KPI SCAN Methodology

This study is based upon the **KPI SCAN®** methodology which analyses the actual usage of SAP systems by end-users. By processing standard historical system data problem areas are identified. Potential improvements in systems operations and business processes are recommended and evaluated. Participating companies extracted the required data from their systems in approximately 30 minutes and forwarded it by email to West Trax for off-site analysis. No software was installed on the clients' systems and there was no disruption to the production environment.

The data analysis consists of 4 stages:

1. As-Is Analysis
- Costs
- Productivity
- Performance
- Quality
- Industry Benchmarks
- Best Practice



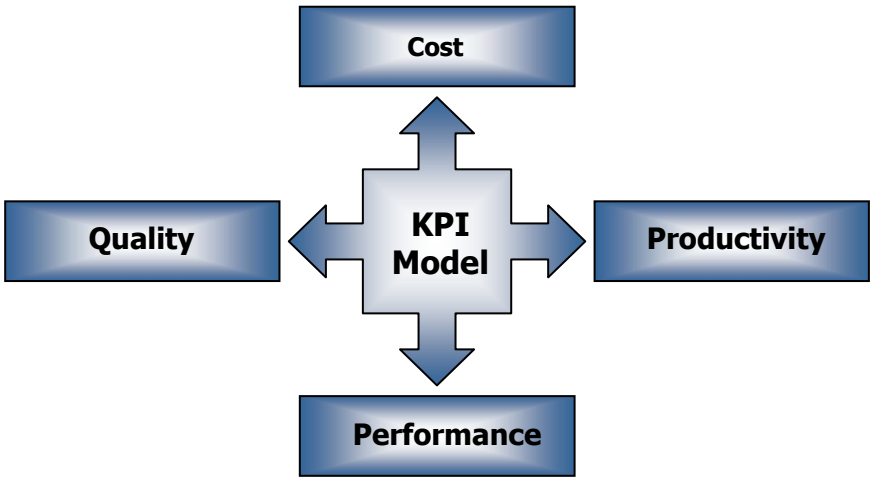
2. Potential
- Cost Savings
- Business Process Optimisation
- Performance Improvements
- Quality Improvements

3. Action Plans
- Cost / Benefit Analysis
- Feasibility Analysis
- Action Selection
- Priority Setting

4. Implementation Strategy
- Timetable
- Resources
- Implementation Plan

ANALYSIS DETAILS

The study analyses categories of Key Performance Indicators (KPI's) to establish the value delivered by the SAP systems. The 4 KPI categories examined are:



Cost KPI's:

- Custom Code Programs
- Unused Custom Code Programs
- Potential Savings
- Maintenance Costs

Productivity KPI's:

- Degree of Standardisation
- Unused Standard Potential
- Supported Industry Sectors
- Supported Core Business Processes
- Unsupported Core Business Processes

Performance KPI's:

- Custom Code Programs
- Standard Transactions

Quality KPI's:

- Usage Frequency
- Background Jobs
- Name Spaces

Cost KPI's

The **Cost KPI section** focuses on Key Performance Indicators impacting the cost of supporting SAP system operations. Custom code programs are examined closely as they play a decisive role in service and maintenance costs.

Generally custom code cannot be revised using standard automated mechanisms. It is written manually using the ABAP language and must be constantly revised to ensure compatability with the current SAP version in use. Time consuming, manual planning and preparation is required. When preparing for SAP release upgrades or consolidations significant incremental financial investment may be necessary.

Even day-to-day use of custom code programs can be very labour intensive, generating increased operational costs. They must be adapted to the support packages (bug fixes etc.) regularly provided by SAP. This process generates an greater workload for the Help Desk than properly documented and tested standard transactions. Training is also more costly and time consuming, as bespoke training programmes must be developed, documented and delivered by the customer.

Documentation is a notorious area of weakness in locally developed code due to personnel changes and individual interpretation. Upgrading and testing in this environment can be expensive, time consuming and may result in quality issues such as loss of process integrity. It also increases the risk of data integrity exposures and unauthorised transactions, further complicating the compliance management process.

The use of some custom code programs cannot be avoided since the current SAP release may not provide the required standard functionality. In this case manual support and maintenance is required to optimise the custom code.

Upgrading may present an opportunity to replace many custom programs with newly provided SAP standard functionality. Unfortunately this option to increase the level of standardisation is often overlooked due to time pressure. As a result the expensive, inefficient custom code continues to be supported. The opportunity to remove unused custom code programs is also often missed at upgrade time. The undetected time consuming and costly overhead continues to increase year-on-year. The potential to significantly reduce costs is disregarded.

The following 4 Cost KPI's are analysed to identify the actual use of custom code programs and the associated costs:

- Custom Code Programs
- Unused Custom Code Programs
- Cost Saving Potential
- Maintenance Costs

CUSTOM CODE PROGRAMS KPI

Definition:

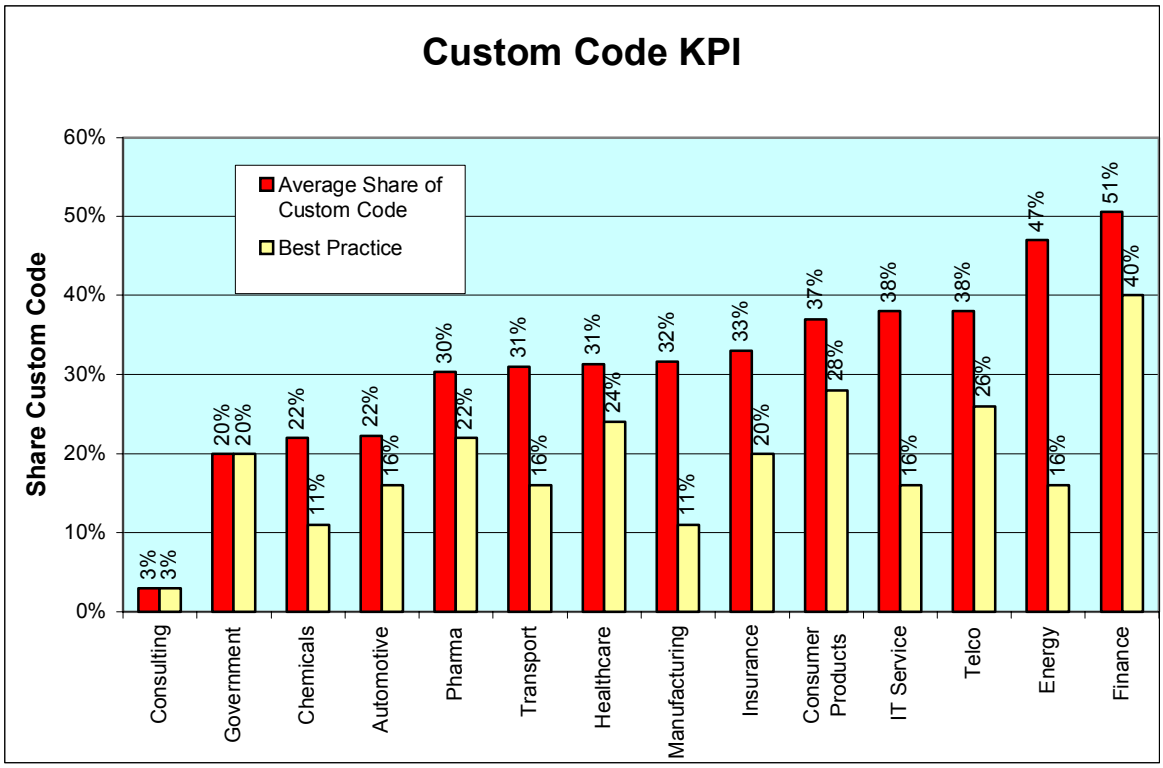
Identify the quantity of custom code programs (Z and Y transactions only) as a percentage of all used programs within the analysed systems.

Calculation:

$$Custom\_Code\_KPI = \frac{(Used\_Custom\_Code + Unused\_Custom\_Code)}{Total\_No.\_Used\_Programs}$$

Results:

The Custom Code Program KPI indicates the average and Best Practice percentage of custom code programs in the analysed systems, by industry sector. Systems in nearly all industry sectors contain a high proportion of custom code programs. The Best Practice values suggest that operations may be possible using less custom code, especially as the Best Practice systems may also contain considerable unrealised optimisation potential.



UNUSED CUSTOM CODE PROGRAMS KPI

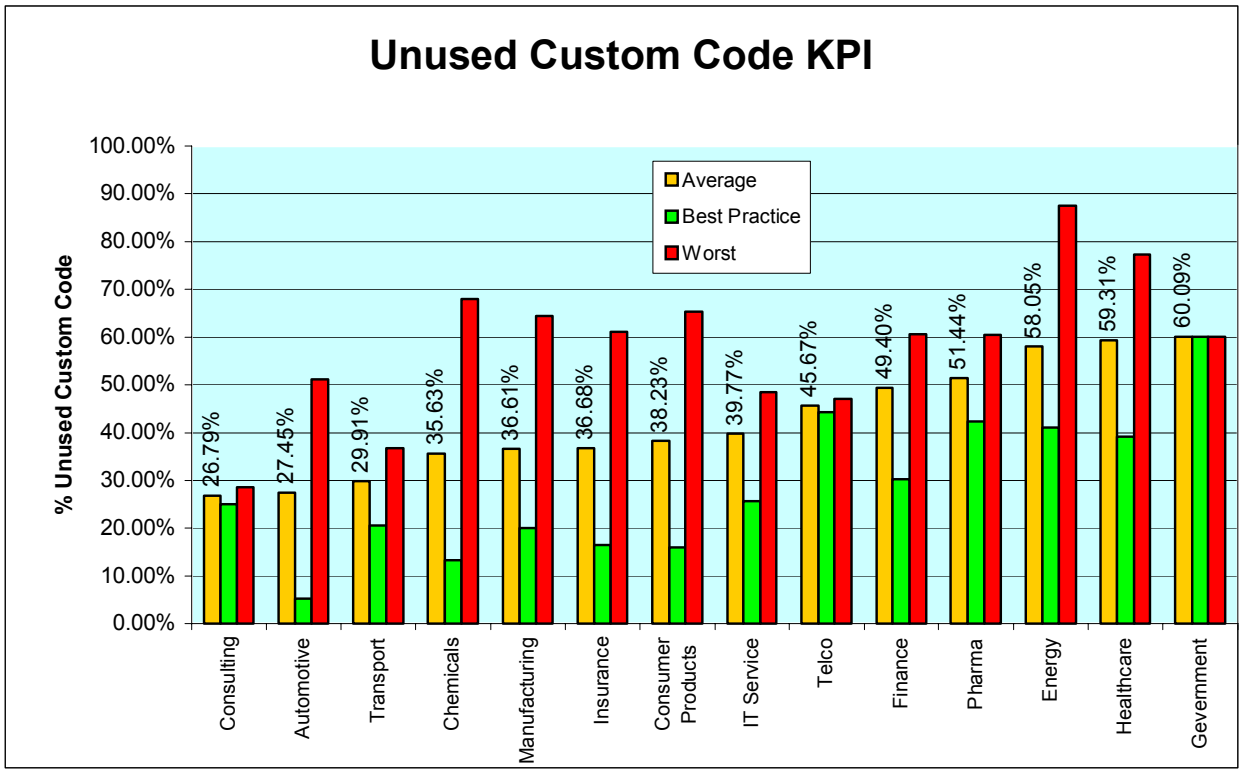
Definition:

Identify the percentage of available custom code programs (Z and Y transactions only) that were unused in the analysed systems.

Calculation:

$$Unused\_Custom\_Code\_KPI = \frac{Total\_No.\_Unused\_Custom\_Code\_Progs.}{Total\_No.\_Custom\_Code\_Progs.}$$

Results:



The **Unused Custom Code Programs** KPI shows the average percentage of custom code programs that were not used during the period analysed. The results show the average by industry sector and Best Practice and Worst Case in each.

The analysis indicates that every system contains unused custom code programs. When viewed as absolute numbers it is apparent that many systems contain hundreds of unused custom programs. These have been left in place due to a lack of transparency or through the implementation of release upgrades on purely technical grounds.

New functionality provided in upgrades may not have been recognised or may have been ignored. This results in many redundant custom code programs remaining in the system, generating increasing overhead year-on-year. As a result every new project will require lengthy and expensive adaptation of the code to the new release.

Understanding which custom code programs are used, and which are not, provides the opportunity to eliminate as much redundancy and overhead as possible during a release upgrade. Failure to take full advantage of this will have a critical adverse impact on the ongoing system operational costs,

Many companies maintain custom code programs which have not been used for months or even years. There may be considerable potential for savings as the following KPI shows.

POTENTIAL SAVINGS KPI

Definition:

Identify the potential to reduce annual maintenance costs by the elimination of redundant custom code programs (Z and Y transactions only).

Calculation:

$$Potential\_Savings = No.\_Unnecessary\_Custom\_Code * Annual\_Workload * Daily\_Rate$$

Units:

- Annual maintenance workload: annual custom code program service and maintenance workload calculated in man days.
- Daily rate: company specific internal cost rate calculated in Euros

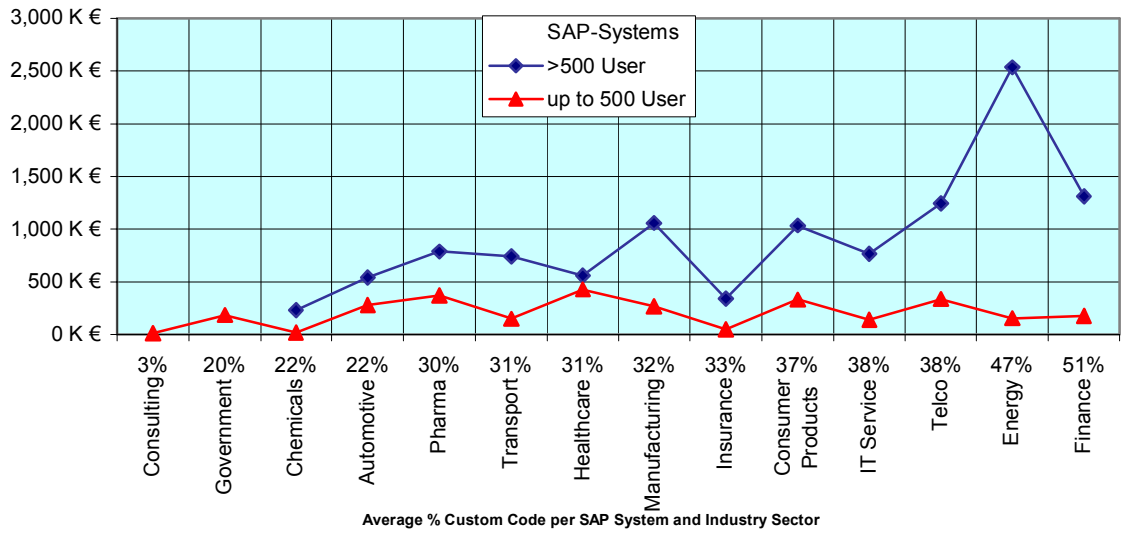
Results:

The **Potential Savings** KPI is influenced by the complexities of individual systems. The red line indicates the average annual potential savings, by industry sector, for systems with less than 500 concurrent users. The blue line plots the results for systems with more than 500 concurrent users.

The potential savings are equivalent to the maintenance costs for redundant custom code programs. These may include: -

- unused Z and Y transactions which are administered and maintained but not used by the end-users
- custom code programs that could be replaced by SAP standard transactions.

Potential Savings KPI



These results clearly illustrate the significant increase in costs incurred when supporting redundant custom code programs.

MAINTENANCE COST KPI

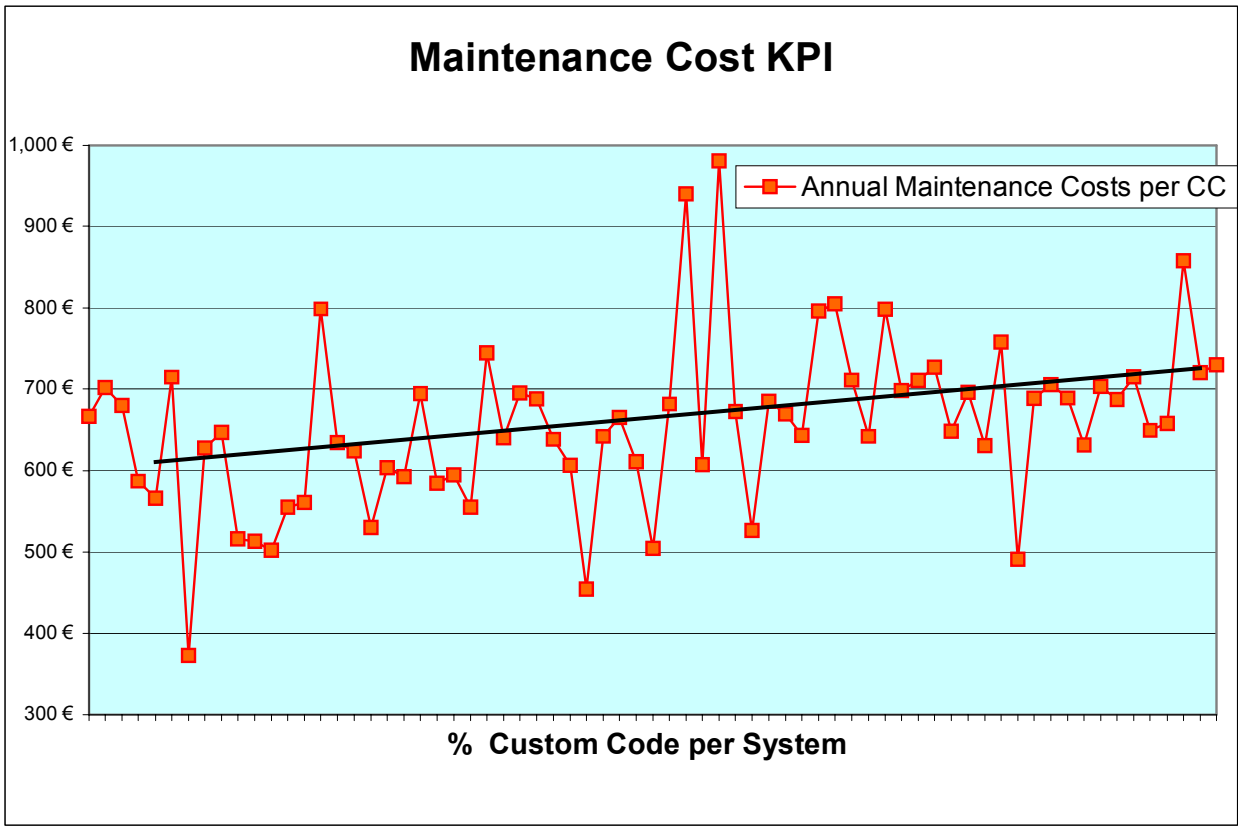
Definition:

Identify the average annual maintenance cost per redundant custom code program (Z and Y transactions only)

Calculation:

$$M\_Cost\_KPI = \frac{Potential\_Savings\_KPI}{Total\_No.\_Unnecessary\_Custom\_Code}$$

Results :



The **Maintenance Costs** KPI illustrates the average annual maintenance cost per unnecessary custom code program in the analysed systems. These may include: -

- unused Z and Y transactions which are administered and maintained but not used by the end-users
- custom code programs that could be replaced by SAP standard transactions.

Nearly all the companies in this study did not have access to accurate custom code program usage data for their systems. As a consequence, wasteful, costly and time consuming maintenance was performed on redundant programs, in addition to the essential work to support productive used programs. This results in the generation of steadily increasing cost overheads year-on-year.

Examples of unused custom code program maintenance actions: -

- Add new query fields in the GUI and extend the "select" for tables
- Insert / change authorisation checks for programs where objects have not been used to date
- Change authorisation objects in order to check new properties (cost centre, cost type)
- Objects must be regenerated and tables replaced when implementing SAP patches
- Revision of routines to reduce processing time
- Change listings (extensions, interactive layouts etc.)
- Revise source code parameters that result from customising and are permanently changed
- Implement hot / support packages to remove high severity software errors

PRODUCTIVITY KPI'S

The **Productivity** KPI's section analyses Key Performance Indicators having an impact on the productivity of SAP software in support of the business. A SAP system runs best when using the available standard functionality as optimally as possible. Unique customised solutions should only be deployed to gain a definite competitive advantage or when the available standard functionality cannot support the requirements of the business. The following KPI's quantify factors impacting system productivity: -

- Percent standardisation
- Unused standard potential
- Unsupported enterprise areas
- Supported core business processes
- Unsupported core business processes

PERCENT STANDARDISATION KPI

Definition:

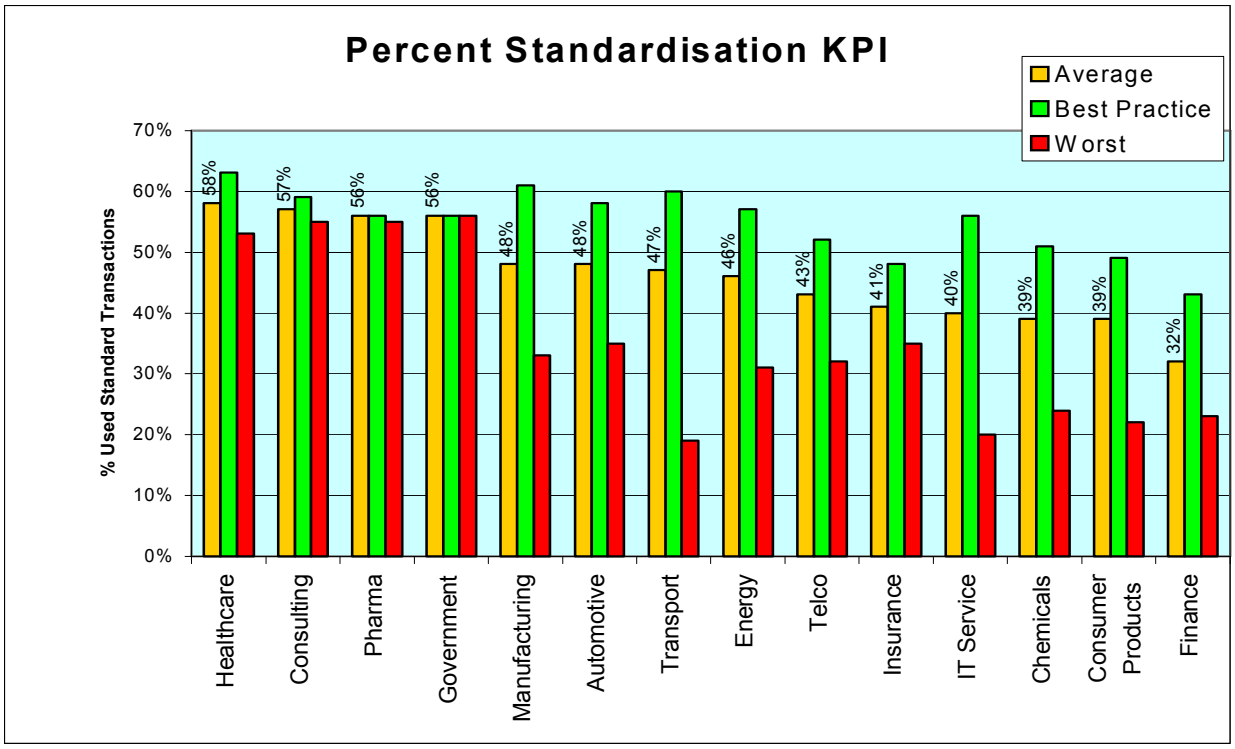
Identify the standard transactions used (S-transactions) as a percentage of all programs used in the analysed systems:

Calculation:

$$S - Percent = \frac{Total\_No.(Used\_S - Transactions)}{Total\_No.\_Used\_programs}$$

Results:

The Percent Standardisation KPI indicates, as a percentage, the average use of standard transactions in the analysed systems, by industry sector. It also shows Best Practice and Worst Case for each.



Most systems analysed in this study show an alarmingly low level of standardisation. This may be due to the advantages of a standard software solution being overlooked when approving the extensive modification of standard code, to support existing business processes, or in response to individual requirements.

Decisions to select standard or custom code solutions are often based on political criteria rather than technical or economical ones. In many organisations harmonisation of the business and the SAP business processes has never been completed. Instead existing business processes are adapted by creating new custom code applications. This seriously impacts the development of integrated business data and the attainment of the expected ROI and business benefits from the SAP installation.

Many country units and subsidiaries try to avoid central control, fearing it is a first step to global restructuring and loss of local control. They may try to protect their autonomy through unique requirements in the form of expensive custom code.

A KPI Scan® analysis will quickly expose many hidden costs associated with politically motivated decisions. Whilst the reduction in maintenance costs is of course a welcome benefit, increasing system standardisation also results in many additional business advantages such as reduced project times, faster adoption of processes and improved data analysis as a result of increased system integration.

UNUSED STANDARD POTENTIAL KPI (S-POTENTIAL)

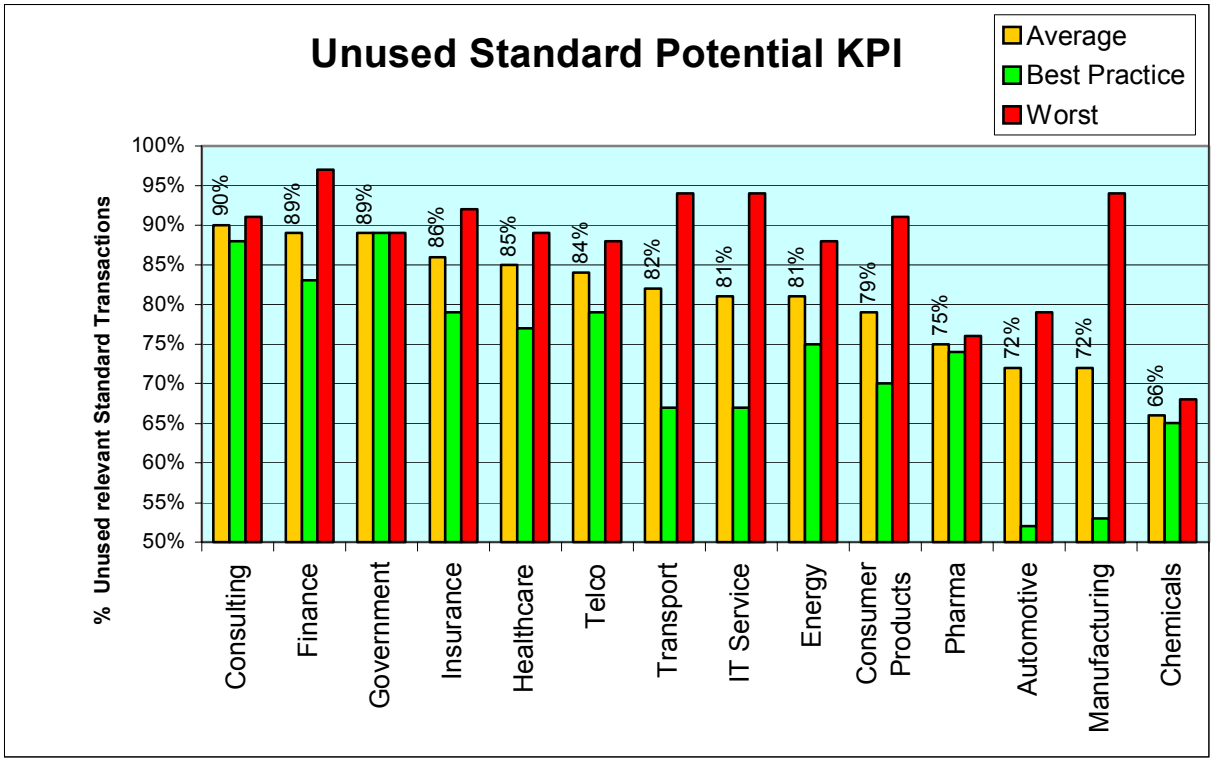
Definition:

Identify the unused relevant standard transactions (S-transactions) as a percentage of the total number of relevant transactions available in the analysed systems.

Calculation:

$$Unused\_S-Potential = 1 - \frac{Total\_No.(Used\_S-Transactions)}{Total\_No\_Relevant\_Available\_S-Transactions}$$

Results:



The **Unused Standard Potential** KPI identifies the average percentage of unused, but company relevant, standard transactions available in the analysed systems. Sector average, Best Practice and Worst Case are shown.

Unused but relevant transactions are standard SAP transactions that:-

- provide functionality that is currently implemented in custom code programs that do not offer any added value.
- could improve the supported business processes.
- would avoid media conversions (improving business process consistency and data integrity).

Note that even in the Best Practice systems in the study more than 50% of the available relevant standard transactions were not used.

Many companies are not aware of all available standard SAP functionality. Poor communication between IT and user departments can result in ill-considered decisions to adopt unnecessary customised solutions.

SUPPORTED ENTERPRISE AREAS KPI

Certain SAP functions are more commonly used than others. The following data identifies those Enterprise Areas that are strongly supported in the systems in this study.

Definition:

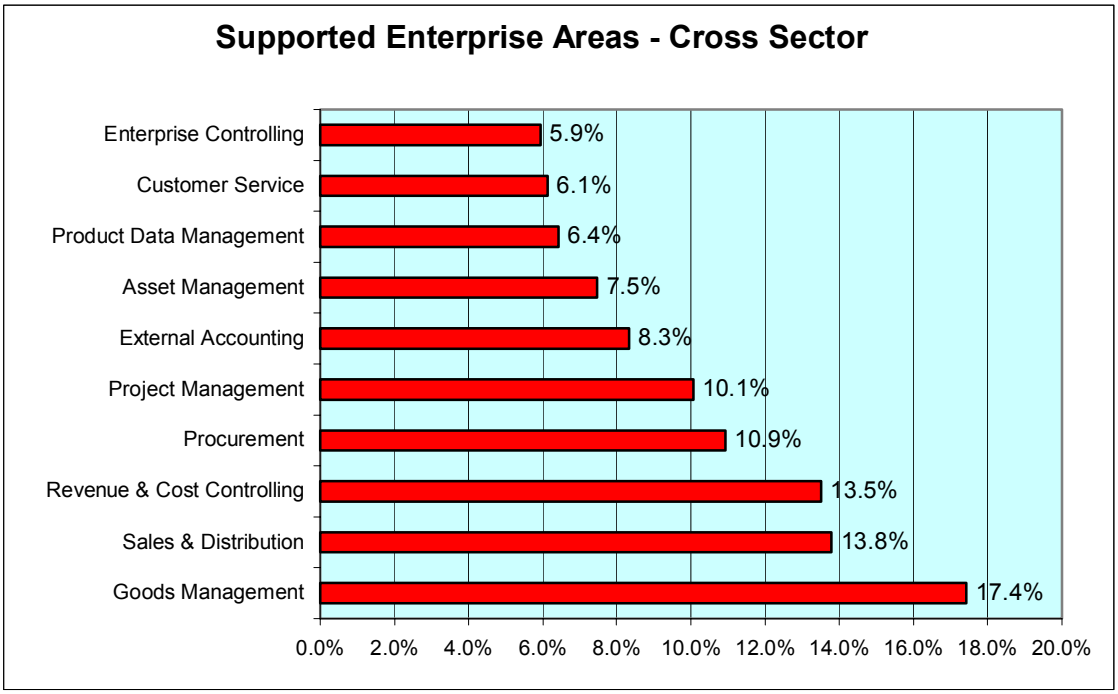
Identify the Enterprise Areas supported by used standard transactions (S-transactions) and the degree to which they are supported.

Calculation:

$$Supported\_Enterprise\_Areas = Mapping(Used\_S-Transactions\_to\_Available\_Enterprise\_Areas)$$

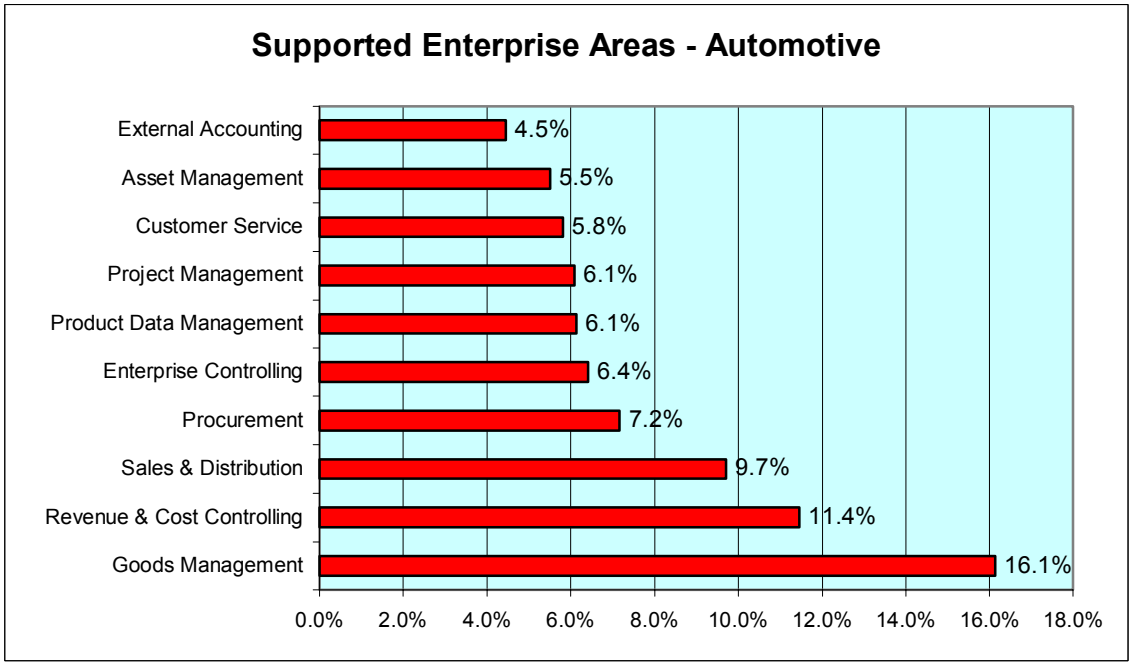
Results:

The **Supported Enterprise Areas** KPI identifies the distribution of all used standard transactions in a system, across the 10 most used Enterprise Areas. The next chart shows the averages across all analysed industry sectors.

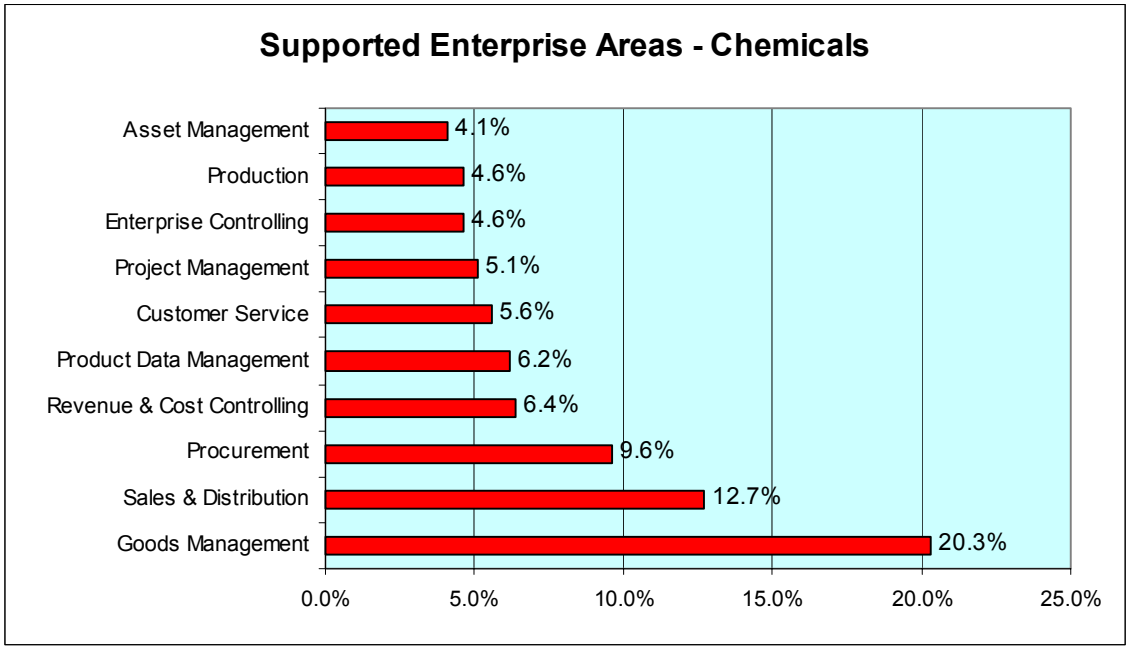


The 10 most frequently supported Enterprise Areas are now analysed by industry, illustrating the degree to which each sector is successful in supporting business processes with SAP standard software.

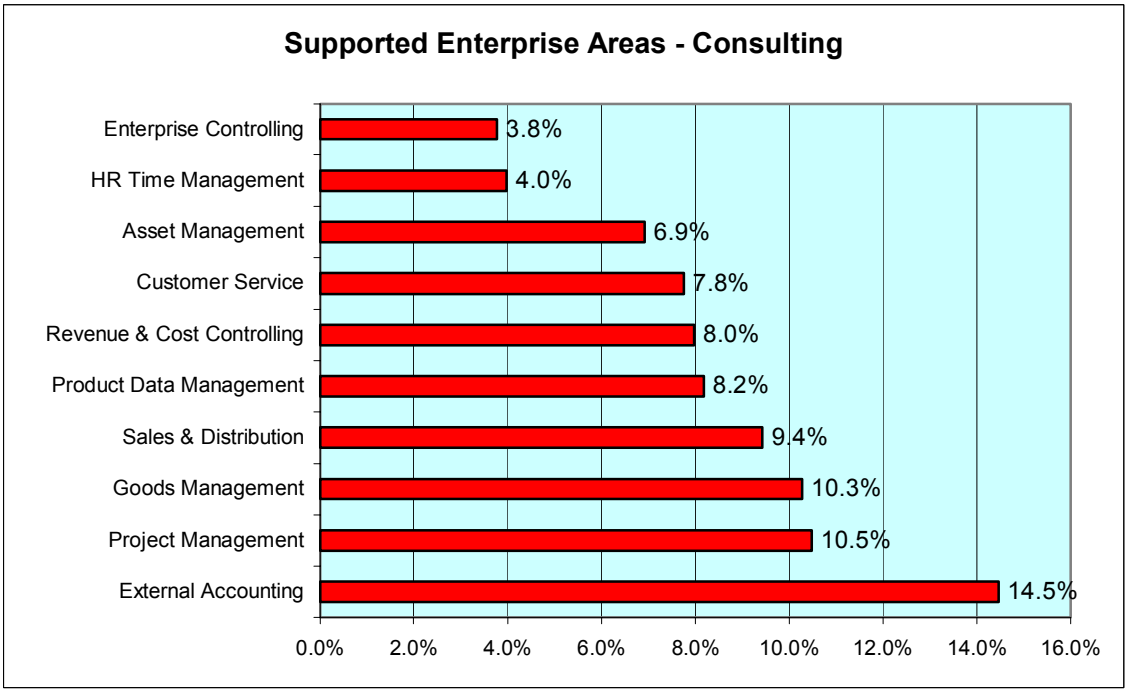
Automotive:



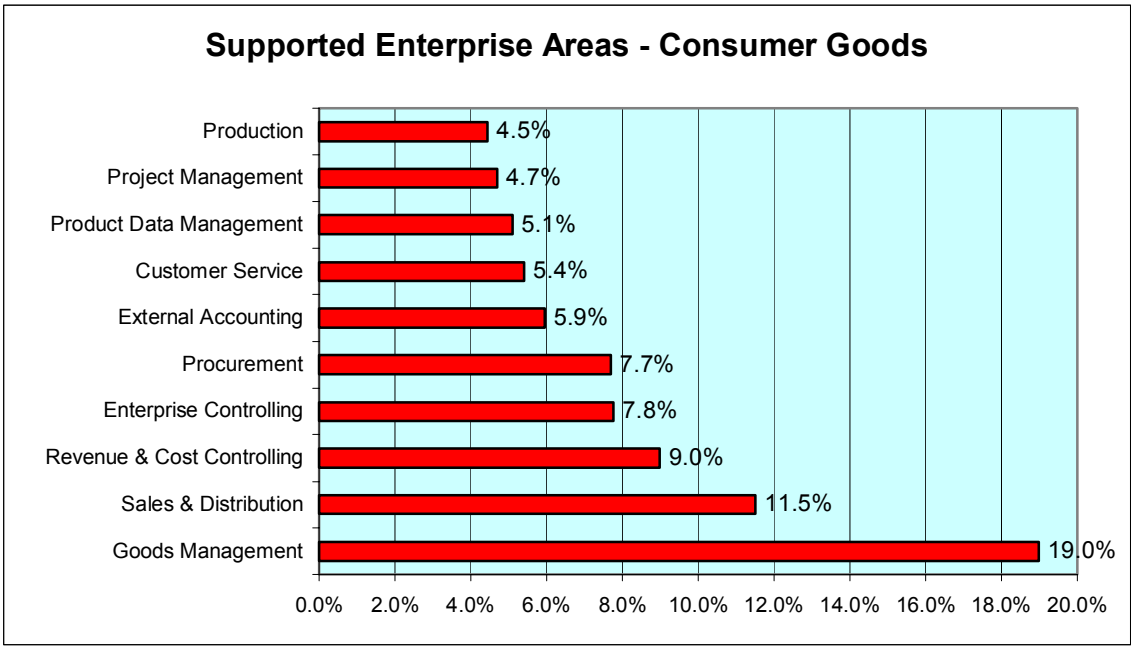
Chemicals:



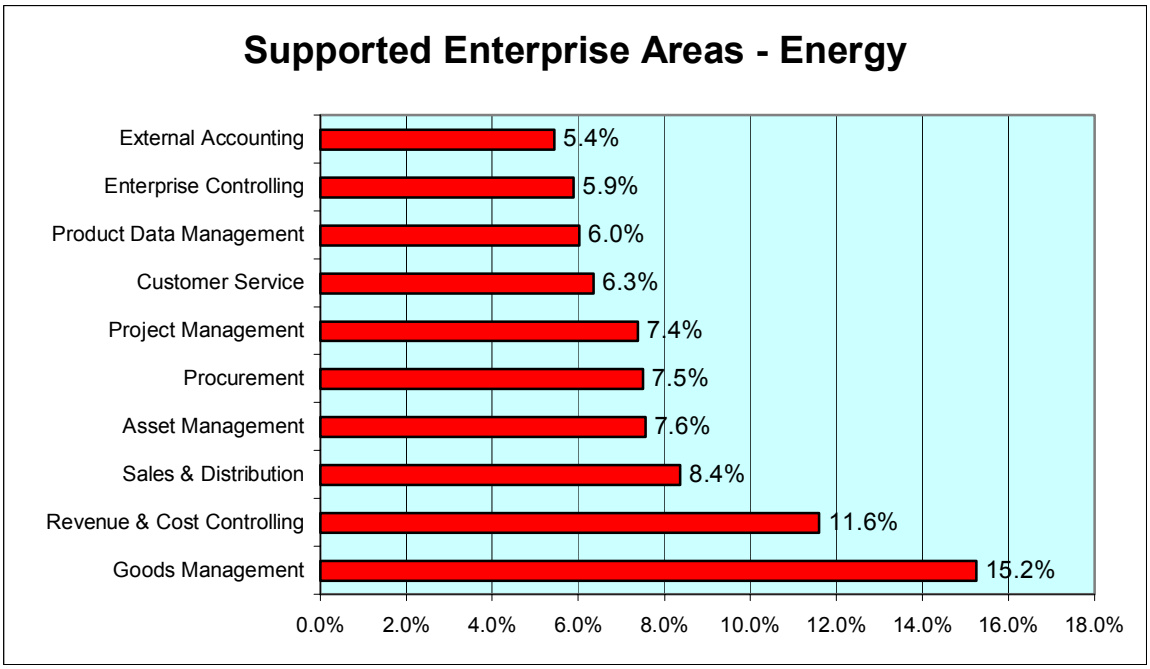
Consulting:



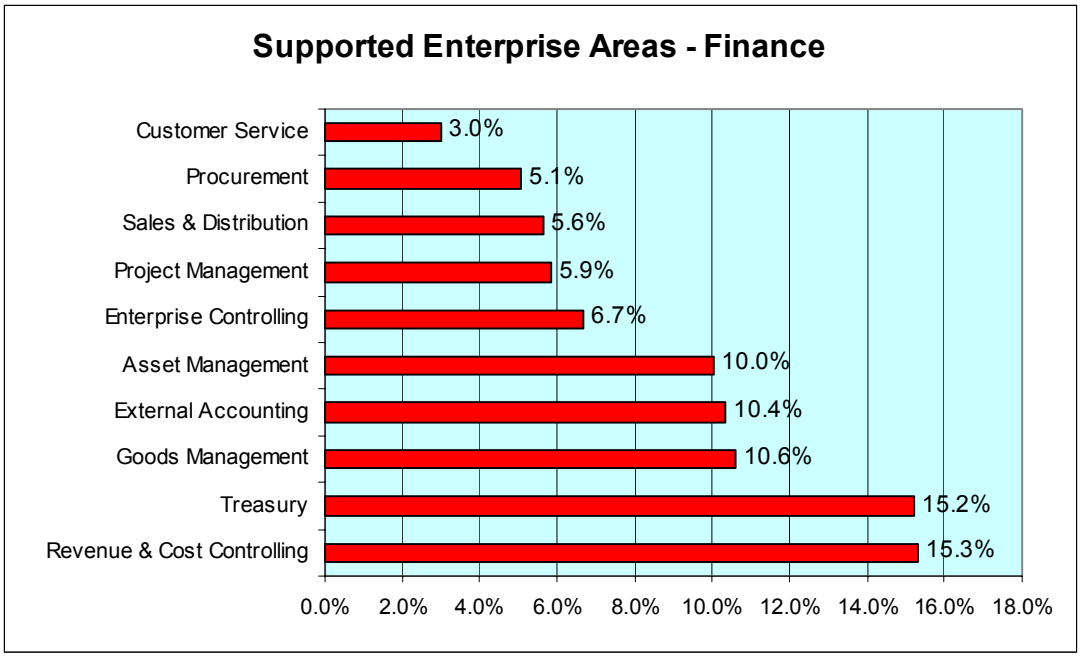
Consumer Products:



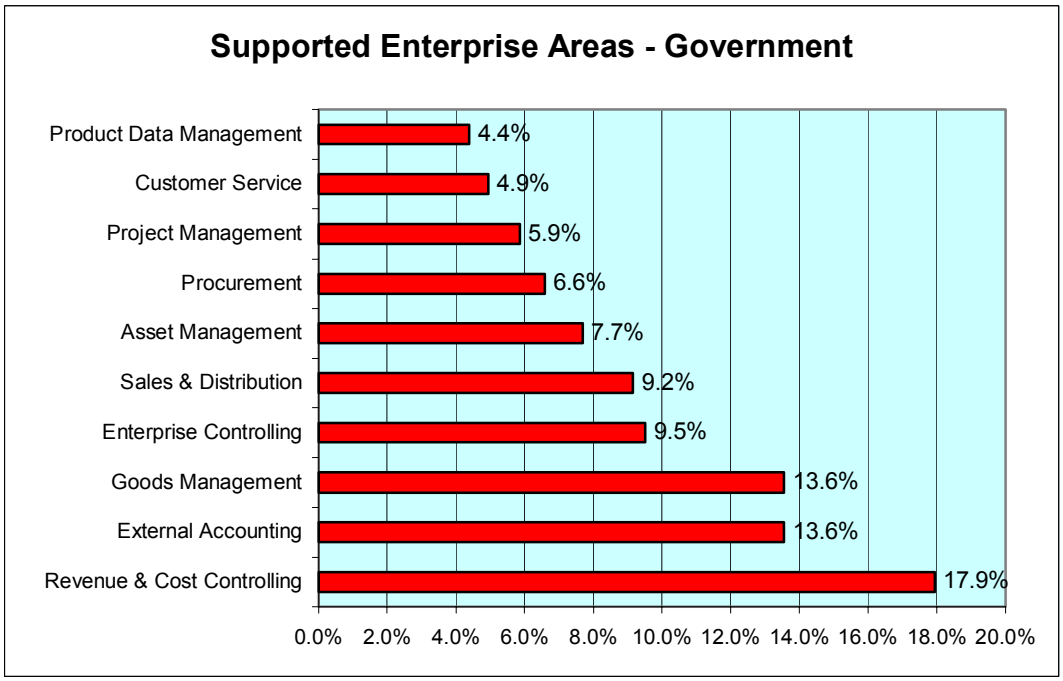
Energy:



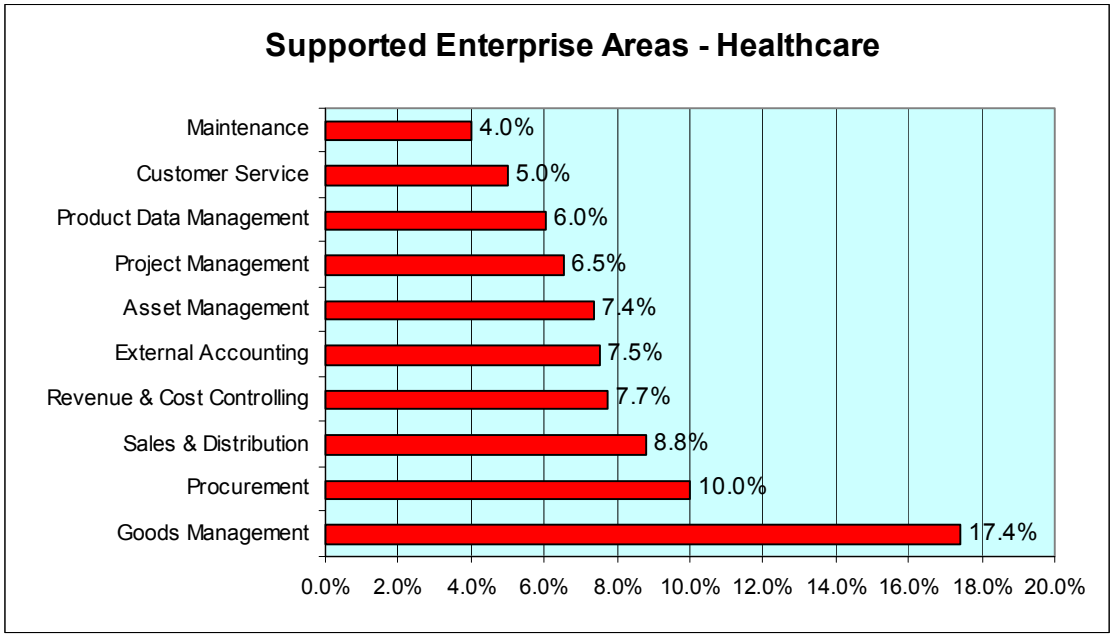
Finance:



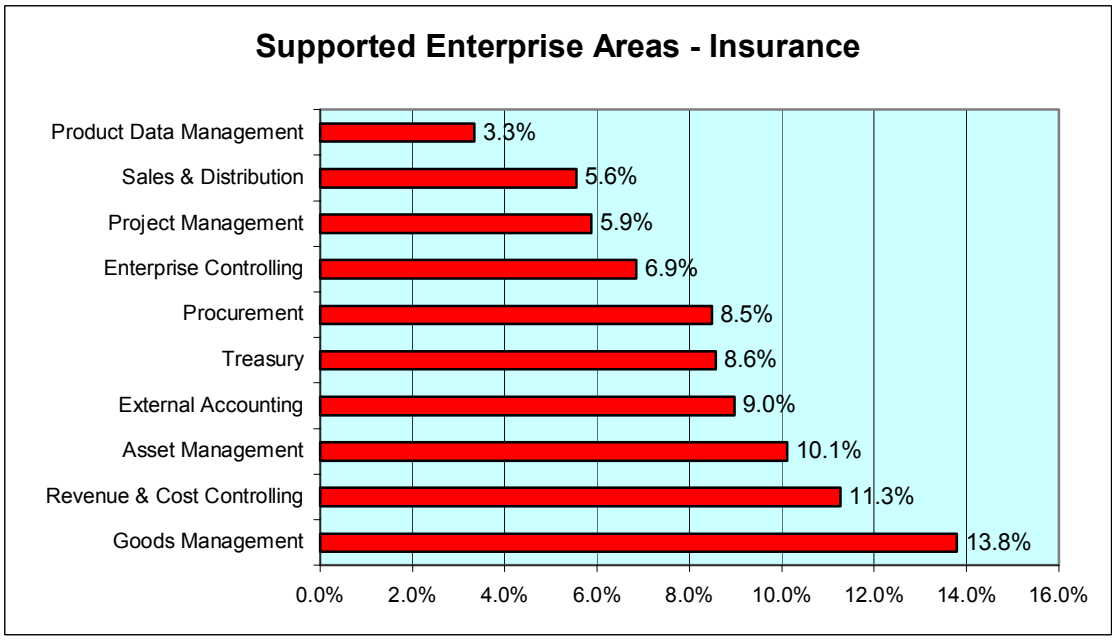
Government:



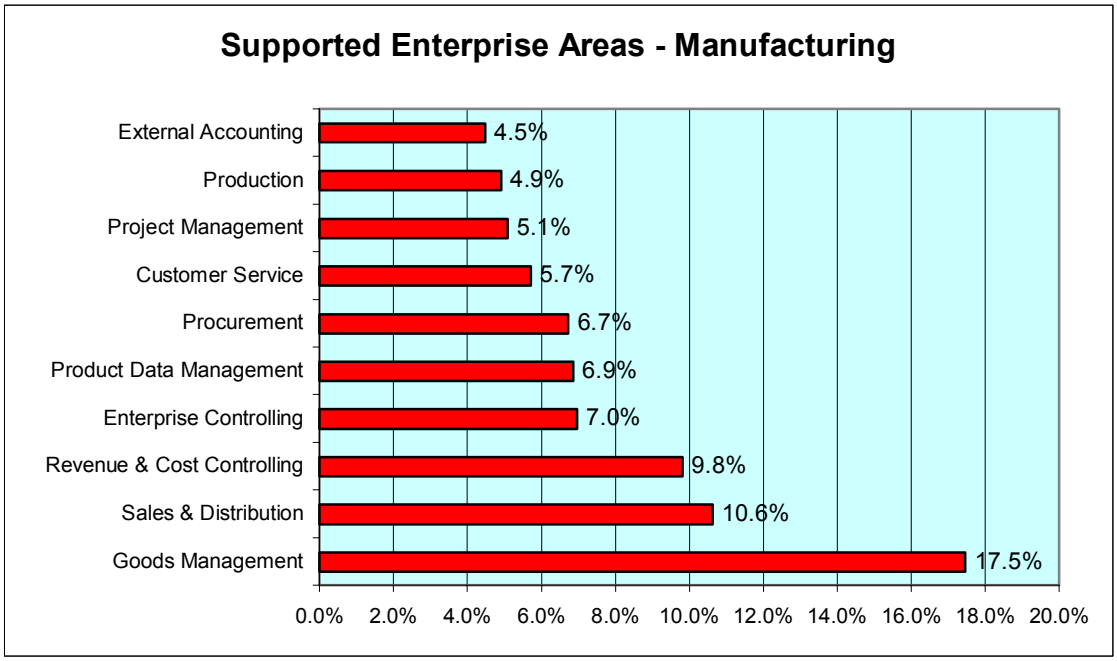
Health Care:



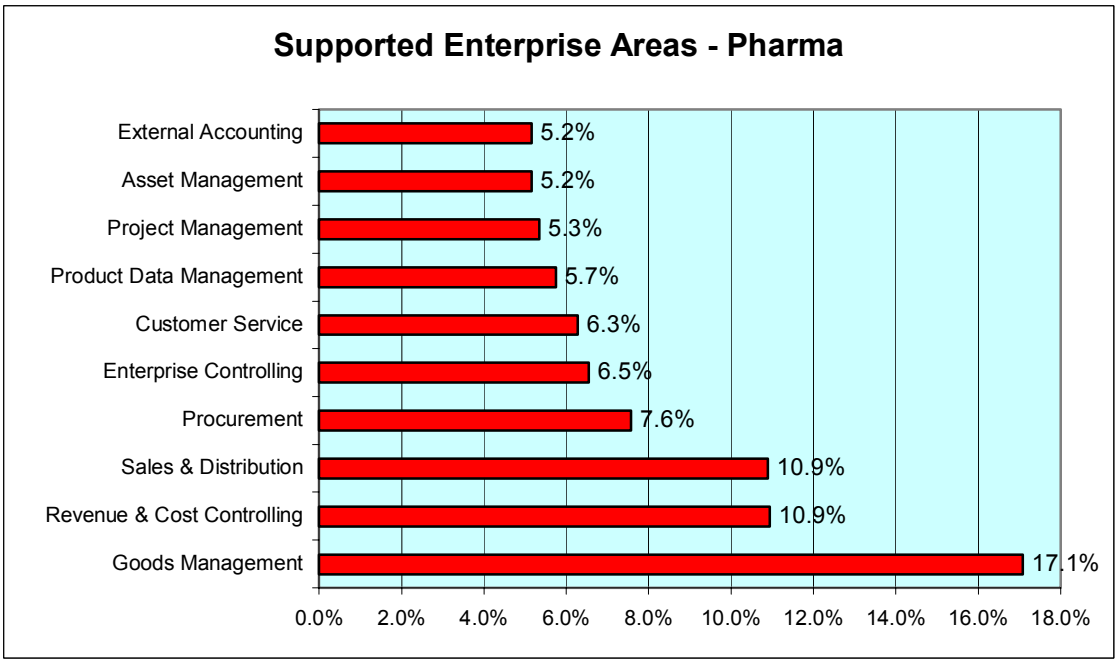
Insurance:



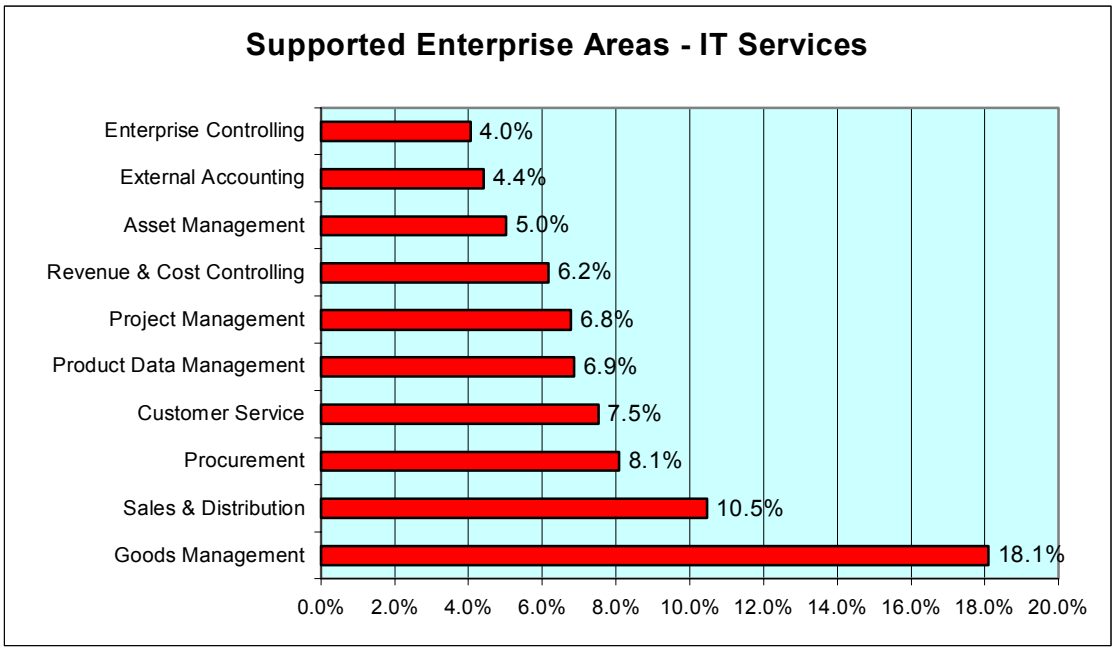
Manufacturing:



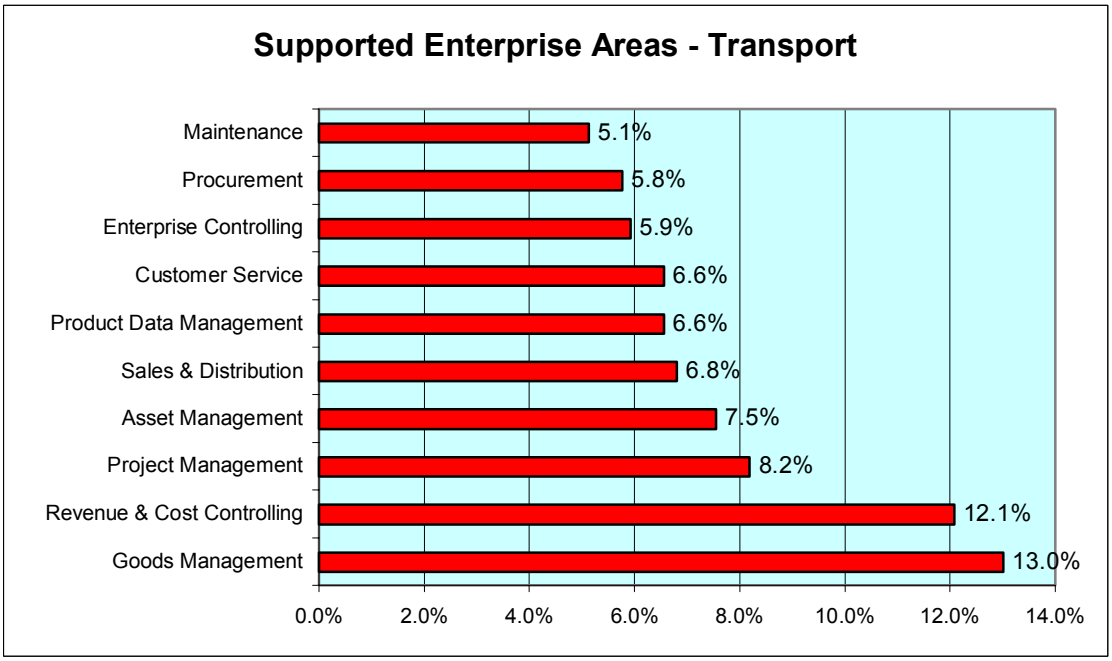
Pharma:



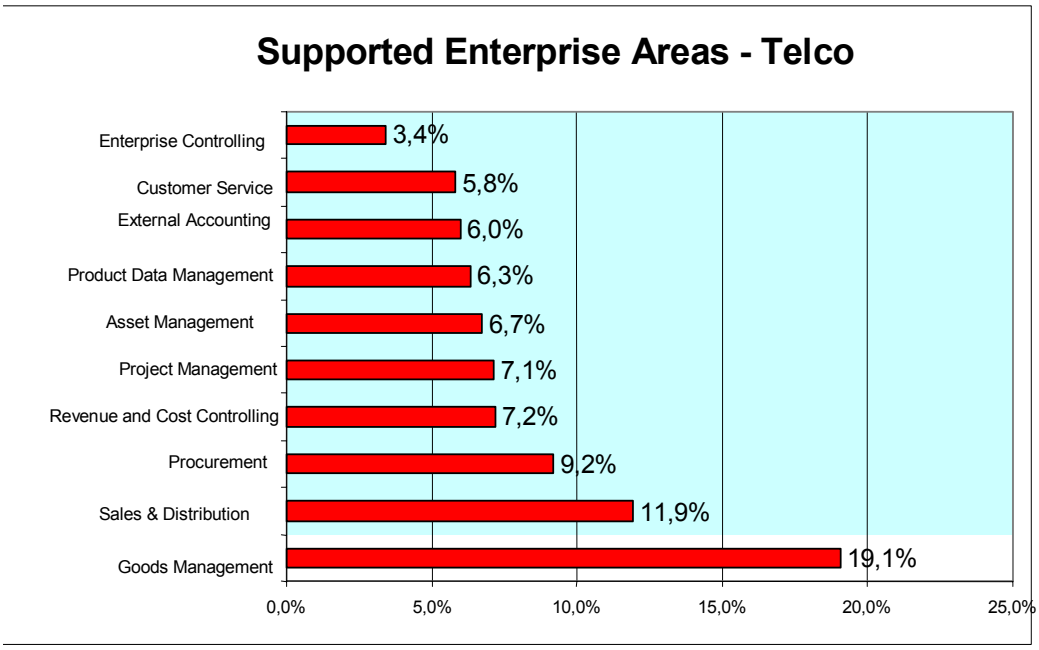
IT Services:



Transport:



Telco:



SUPPORTED CORE BUSINESS PROCESSES KPI

Every Enterprise Area features specific Core Business Processes. In every SAP system it is possible to identify those processes which are best supported by the standard functionality in the Enterprise Areas. This KPI identifies the Core Business Processes occurring most frequently in all the analysed systems.

Definition:

Identify the Core Business Processes which are supported by used standard transactions (S-transactions)

Calculation:

$$Supported\_Core\_Business\_Processes = Mapping(Used\_S-Transactions\_To\_Available\_Core\_Business\_Processes)$$

Results:

The Supported Core Business Processes KPI concentrates on the three most supported Enterprise Areas identified in the cross sector chart on page 22.

Intensively supported Core Business Processes by Enterprise Area:

Goods Management

1. Logistics
2. Master data
3. Purchase
4. Sales

Sales & Distribution

1. Sales order processing
2. Third-party order processing
3. Management of consignment stocks
4. Direct sales order processing

Revenue and Cost Controlling

1. Period Closure
2. Result and cost planning
3. Cost/benefit analysis

UNSUPPORTED CORE BUSINESS PROCESSES KPI

In addition to identifying Core Business Processes that are well supported by SAP standard software, the study also identifies those that are not. In many cases the Core Business Processes may be relevant to the business, but they are not supported by the use of available standard functionality. The reasons for this vary by organisation. For example: -

- The company is not aware that a Core Business Process could be supported by available standard functionality. Expensive custom code is used instead.
- The standard functionality does not meet the specific requirements of the business.
- The standard functionality meets the requirements but a custom code solution is accepted for political or convenience reasons.

Definition:

Identify the unsupported Core Business Processes in the relevant Enterprise Areas.

Calculation:

$$UnSupported\_Core\_Business\_Processes = Available\_Core\_Business\_Processes - Supported\_Core\_Business\_Processes\_KPI$$

Results:

The **Unsupported Core Business Processes** KPI focuses on the most supported Enterprise Area identified in the cross sector chart on page 22.

Unsupported core business processes in the **Goods Management** Enterprise Area**:**

1. Subsidiary network
2. Cross-functional processes (i.e. agency business)
3. Strategic sales (i.e. sales actions, seasonal sales, online store)

Although the Goods Management Enterprise Area is the one most supported across all sectors, none of the systems analysed in the study support the above 3 Core Business Processes with the available standard functionality.

PERFORMANCE KPI'S

The **Performance** KPI section identifies Key Performance Indicators providing response time data for programs used in the analysed systems. The study compares the average response times of applications, by category, using a reference value of one minute. The following KPI's are identified: -

- Custom code program performance
- Standard transaction performance

CUSTOM CODE PROGRAM PERFORMANCE KPI

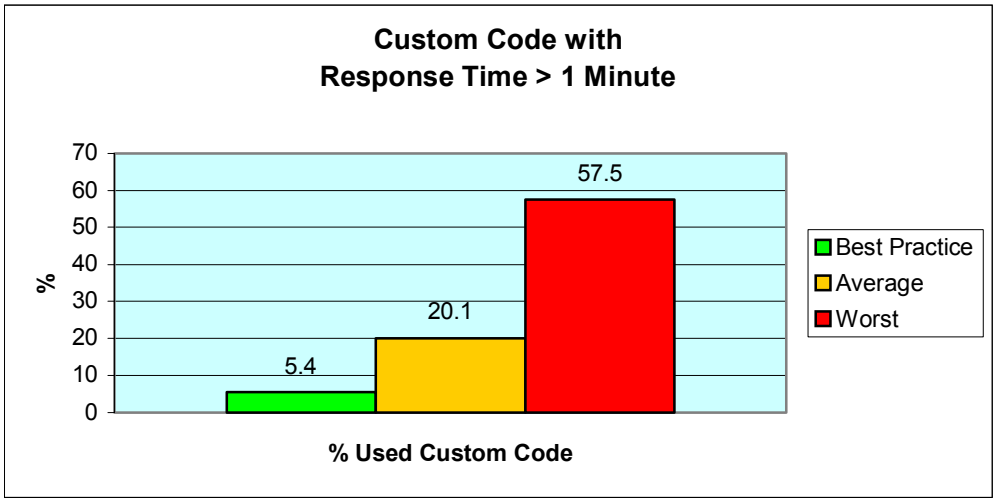
Definition:

Identify the percentage of used custom code programs (Z and Y transactions only) with an average response time of over 1 minute.

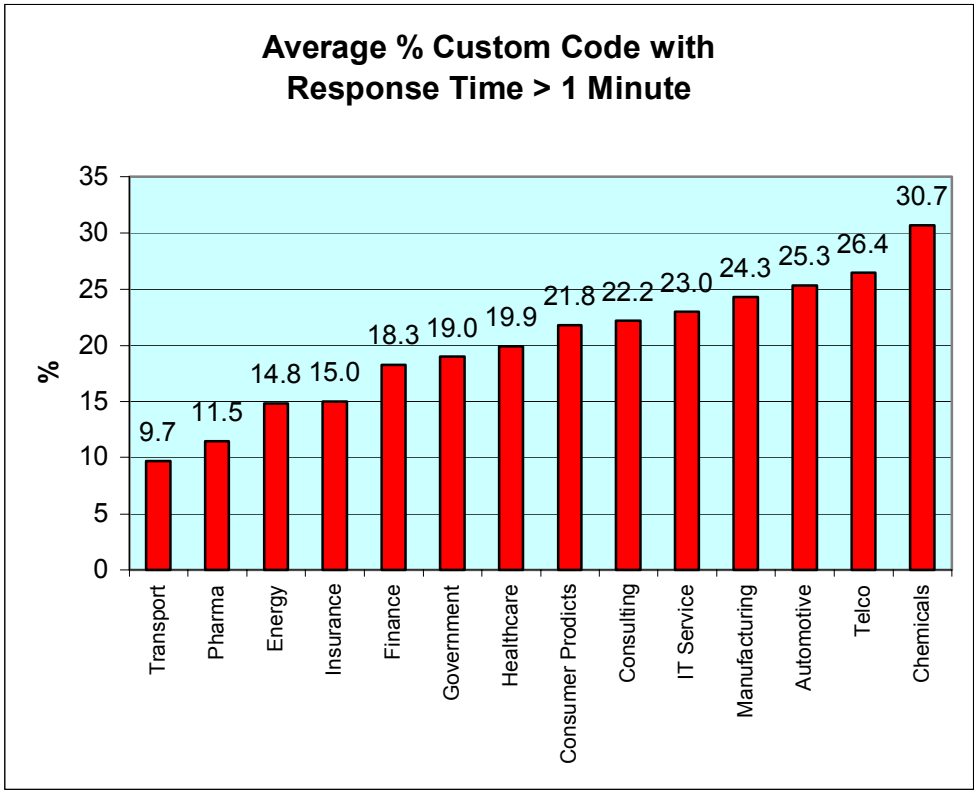
Calculation:

$$Performance\_Custom\_Code = \frac{Total\_No.(Custom\_Code\_with\_RT > 1\_Minute)}{Total\_No.\_Used\_Custom\_Code}$$

Results:



The **Custom Code Program Performance KPI** results across sectors are shown in the above chart and by sector in the chart below.



The majority of custom code programs analysed showed very poor response times. Companies will argue that this is due to factors such as reports running overnight as background jobs. Optimising the code is therefore not considered important. However, spot checks carried out during the analysis show that these programs often had an adverse impact during the working day, when they are not used exclusively as background jobs. Some of them even proved to be dialog programs, causing inevitable user dissatisfaction and loss of productivity.

As organisations extend and consolidate their SAP landscapes, across the enterprise and beyond, custom code programs producing poor response times become more problematic. In a 24 hour, round the world operating environment poor performance may impact productivity and user satisfaction for someone, somewhere, at any time of day. In the extended enterprise environment this could include customers and suppliers.

The next chapter shows that in most cases standard transactions offer much better performance. The results confirm the view that custom code programs should only be used when suitable standard code is not available. Where it is essential to use custom programs they should be subject to regular quality checks to ensure performance is optimised and core business processes are appropriately supported.

STANDARD TRANSACTION PERFORMANCE KPI

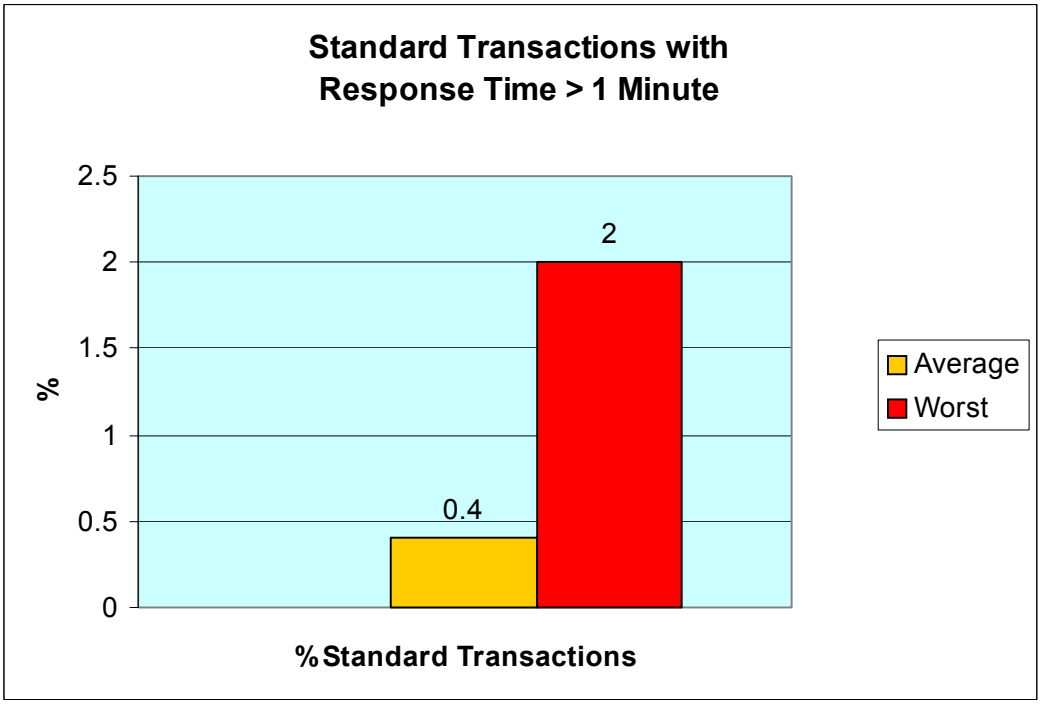
Definition:

Identify the percentage of used SAP standard transactions (S-transactions) with an average response time of more than 1 minute.

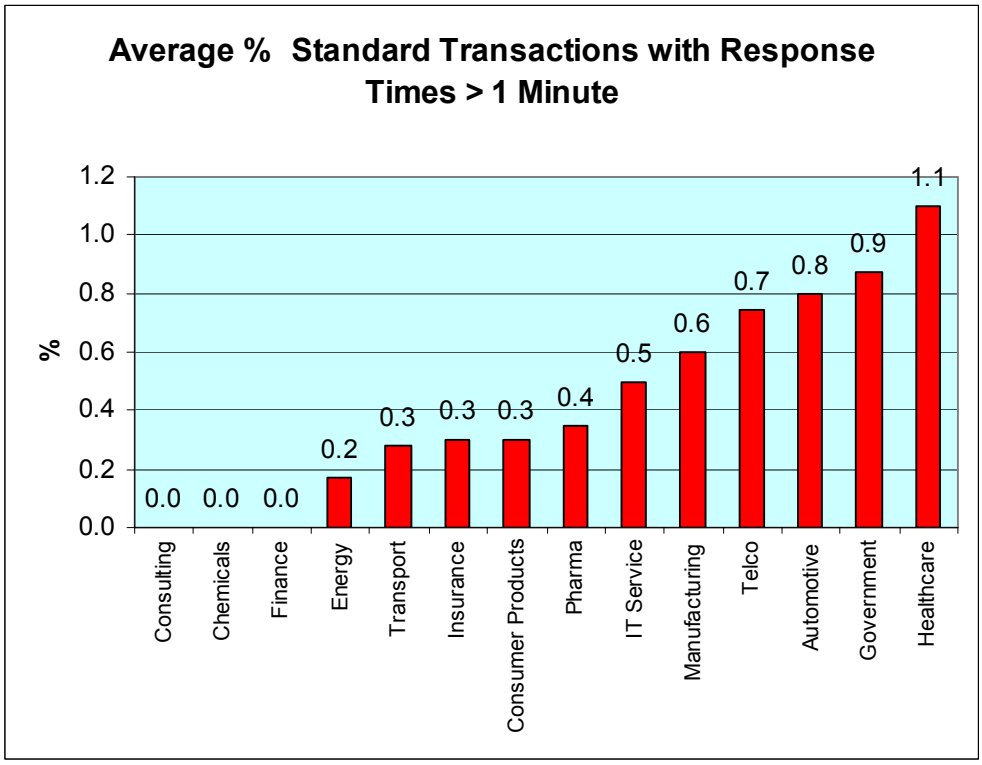
Calculation:

$$Performance\_S-Transactions = \frac{Total\_No.(S-Transactions\_with\_RT>1\_Minute)}{Total\_No\_Used\_S-Transactions}$$

Results:



The Standard Transaction Performance KPI was analysed across all sectors (see diagram above) and by sector (see diagram below).



A comparison of standard transaction performance with that for custom code programs clearly shows that standard code consistently produces higher performance. There are cases where standard transaction performance is unacceptable and as a result companies may need to create their own custom code programs to improve the situation but this should be an exception.

SAP standard transactions are subject to ongoing quality checks and have been tested by SAP and the users of the many companies using the software. However, custom code programs are often subject of poor quality checks in individual companies and are tested less rigorously than standard programs. Rotating responsibilities and improper documentation significantly complicate custom code quality management.

Many companies allow the replacement of standard transactions with custom programs in order to satisfy end user's requirements. These decisions are often taken without a thorough analysis of the long-term consequences.

Please note that in almost all the systems analysed in this study the used custom code programs were mostly reports. The used standard transactions were mostly dialog programs. It is important to be aware of this when interpreting the Performance KPI results.

QUALITY KPI'S

The **Quality** KPI section analyses factors impacting system quality. The KPI's are as follows:

- Usage Frequency
- Background Jobs
- Name Spaces

USAGE FREQUENCY KPI

Definition:

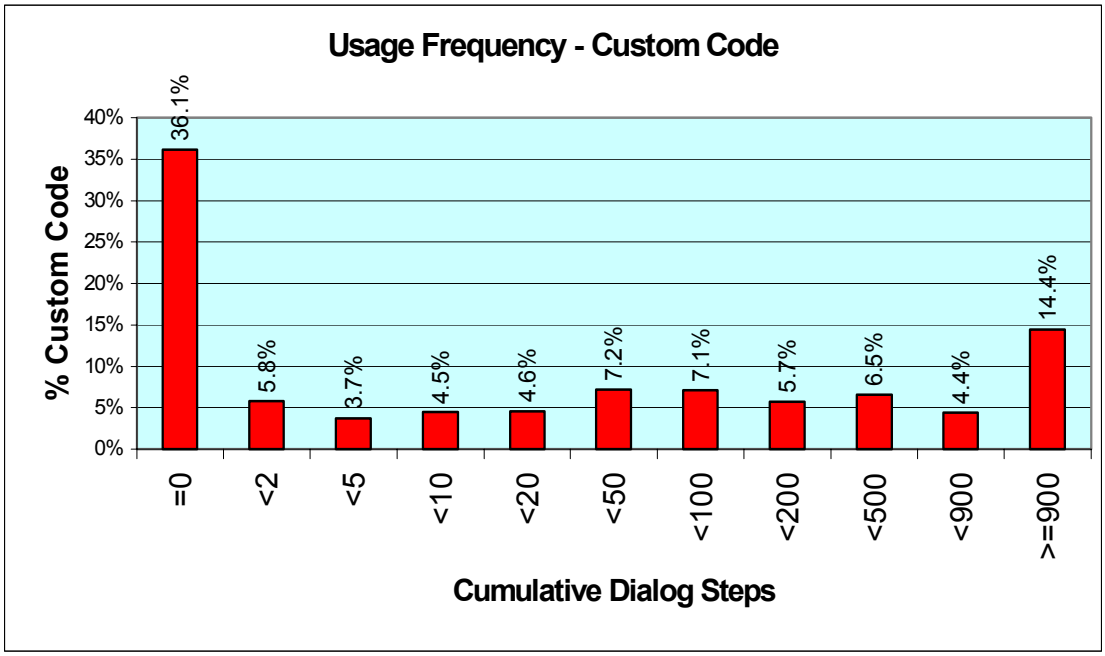
Identify the usage frequency of Custom Code programs used within the analysed time period.

Calculation:

$$Usage\_Frequency = Cumulated\_Dia\log\_Steps\_used\_programs$$

Results:

The **Usage Frequency KPI** identifies the average usage frequency of programs, (this study focuses on Z and Y transactions), in the analysed systems, across all sectors.



Since supporting custom code programs is very time consuming and expensive, companies would be expected to only resort to them where competitiveness can be improved and comparable standard solutions are not available. As previously demonstrated this is not the case. The study shows that most of the analysed systems contain high levels of custom code, but the usage frequency analysis demonstrates that only approx. 25% of the programs are intensively used. Over a third of them are never used.

Increased system transparency in the form of accurate custom code usage data, as provided by a KPI Scan® analysis, allows differentiation between used and unused programs, enabling the elimination of redundant code.

BACKGROUND JOBS KPI

Background Jobs are programs triggered by another program that is not visible to the user. Frequently, a program may trigger more than one background job. Organisations seldom know how many background jobs exist in their systems and which programs trigger which jobs.

The following KPI identifies planning and performance risks: -

Definition:

Identify existing background jobs.

Calculation:

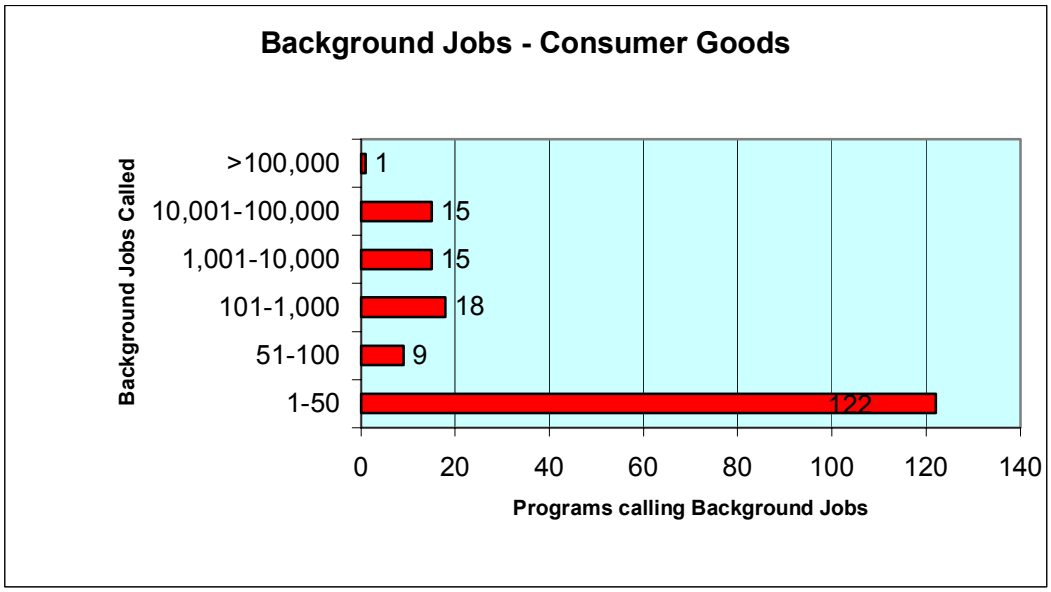
$$Background\_Jobs = Sum(Backgroound\_Jobs\_per\_used\_Program)$$

Results:

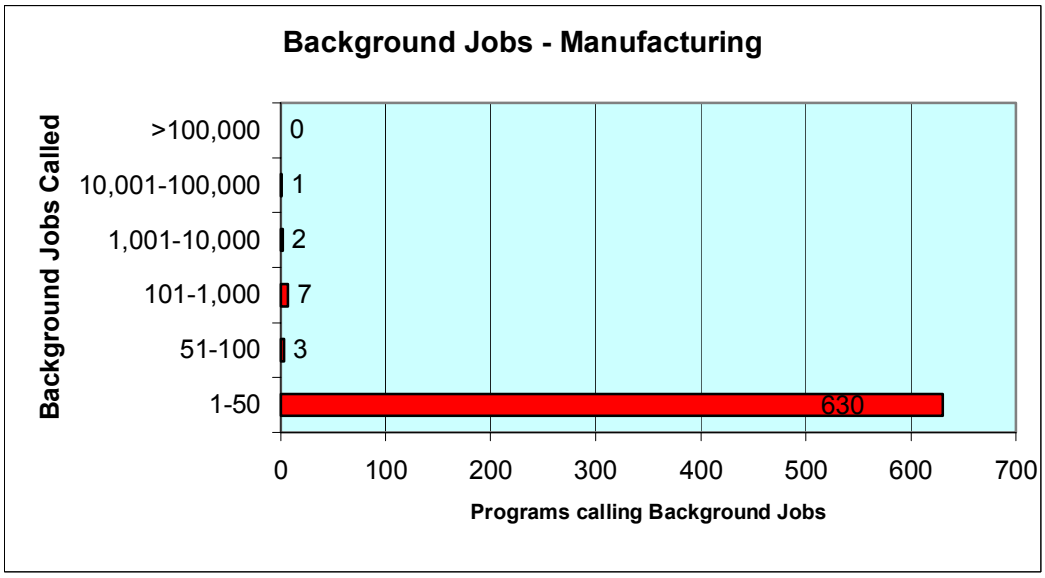
This study analyses 5 systems from different industry sectors and identifies the number of trigger programs calling background jobs in each of them.

Only the trigger programs are taken into account when planning projects such as release upgrades or consolidations. The non-transparent background jobs are often overlooked during project planning and scoping, resulting in the workload involved being seriously under-estimated. Budget overruns and missed project deadlines are the likely consequence.

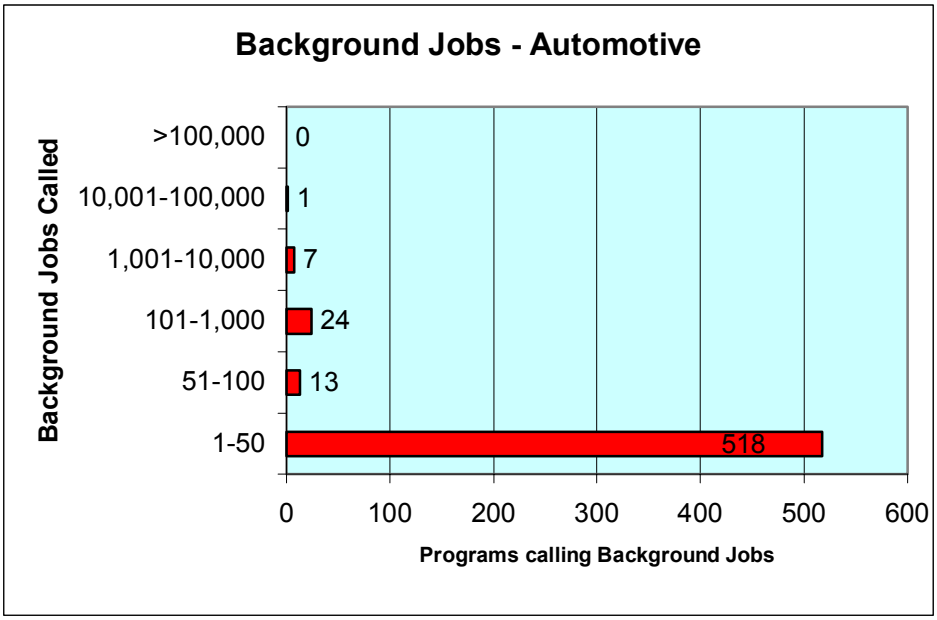
Company 1 – Consumer Goods:



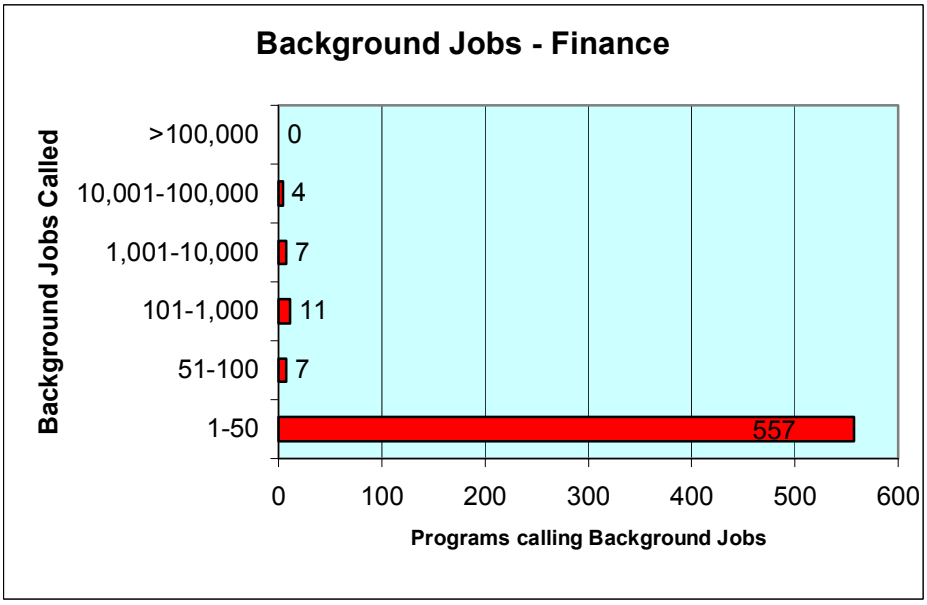
Company 2 - Manufacturing:



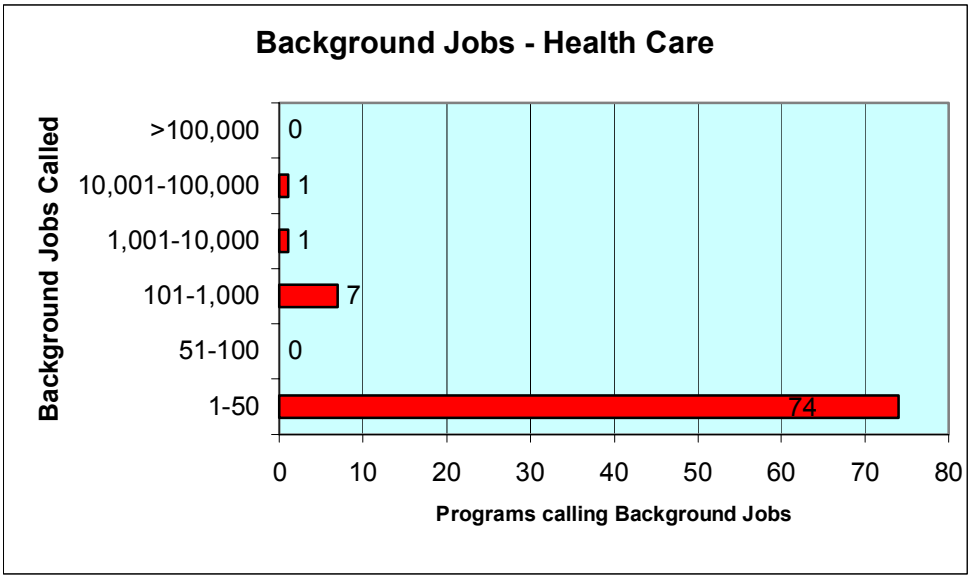
Company 3 - Automotive:



Company 4 - Finance:



Company 5 – Health Care:



NAME SPACES KPI

Special name spaces may be provided for non-standard transactions in SAP systems to keep the system transparent and well structured. For example the Z and Y name spaces are reserved for custom code ABAP programs. Third party suppliers also use individual name spaces to identify their company or product name.

Many organisations create name spaces for specific subsidiaries or locations in order to exercise some control over these business units whilst avoiding too much central influence. This results in unstructured and disorganised custom code development and standard code modifications. Often two locations will develop similar functionality whist unaware of each others efforts. Redundancy and overhead are inevitable, greatly increasing the workload associated with system maintenance and the planning and implementation of release upgrades.

Definition:

Identify the number of programs in the existing name spaces.

Calculation:

$$Name\_Spaces = Sum(Used\_Programs\_per\_Name\_Space)$$

Results:

This study analyses 5 companies from various industry sectors to demonstrate different name space usage.

Company 1 - Manufacturing:

Number of name spaces: 7

Number of programs in the name spaces: 736

Company 2 – Consumer Goods:

Number of name spaces: 3

Number of programs in the name spaces: 46

Company 3 - Automotive:

Number of name spaces: 10

Number of programs in the name spaces: 118

Company 4 - Chemical:

Number of name spaces: 3

Number of programs in the name spaces: 81

Company 5 - Pharma:

Number of name spaces: 3

Number of programs in the name spaces: 17

STUDY SUMMARY

Most of the systems analysed in the study show significant under-utilisation of available major functional components that are incurring SAP license and maintenance charges. Redundant or inefficiently used custom code programs, incompletely utilised standard potential and a lack of awareness of available standard functionality all contribute to the much criticised costs associated with SAP systems operations and projects

A KPI Scan® analysis, as employed in this study, can rapidly provide current, independent data, identifying hidden costs, wastage, inefficiencies and other improvement areas. Whilst reduced maintenance costs are a welcome effect, additional benefits may be gained from improvements in project times, business processes and system consistency. More reliable data helps maintain alignment between the system and the business, increasing competitiveness, and adding value to the business.

The results obtained in this study highlight consistent, major discrepancies between:

- functionality that is available and that which is implemented.
- functionality that is implemented and that which is actually used.

The purpose of a standard software solution is not to harmonise an organisation, as is often assumed, but to simplify and optimise processes and so focus the business on its core competencies, enabling it to remain competitive and successful in the long term.

Many organisations continue to permit extensive customisation of their systems. As a result, many of the benefits and added-value expected from their strategic investment in SAP are lost. The consequences are increased maintenance and project costs, unnecessary media conversions, unapproved workarounds, problem processes, poor performance, lost quality, increased training needs, inaccurate documentation, compliance exposures and user dissatisfaction.

APPENDIX A - BIOGRAPHIES

Diana Bohr

Diana Bohr is co-owner of West Trax Deutschland Ltd.

As Technology Director she is responsible for the development of analysis methods to increase the added value companies derive from using their SAP systems (KPI Scan, Benchmark Express). Under the direction of Diana Bohr the individual analyses are coordinated and respective process models are developed for the customer. She also focuses on SAP process benchmarking, cost/benefit analyses for big German industrial companies as well as IT market research and intelligent data analysis.

Diana Bohr is graduate computer scientist and has well-founded IT knowledge with a special emphasis on business process analyses, modelling and optimisation.