

SAI - POLAND

CAAT - or - jNIK audit electronic papers



Tradition

SAI Poland, the Supreme Chamber of Control (SCC or NIK!) is an organisation with a long tradition. It has been independent from the Government since it was created in early 1919 - just after Poland regained independence as the Second Republic. It was descended from several financial and administrative audit institutions with roots in early 19th century. In the post-war People's Republic, NIK changed its face several times to strengthen its position as indispensable element of modern democratic Poland.

Complex Audit Procedure

Our well-established tradition, which is reflected in current regulations, imposes a specific way of work. Most of our audits are of a multi-location type. A single, and usually very detailed, Audit Plan guides all the audit teams in a co-ordinated audit but every auditor has a large margin of autonomy especially when faced with unexpected findings. Our auditors produce just one working paper (called a Protocol) per location and combine in it everything that has

been produced and collected during the audit. It should be a snapshot of the situation of the auditee.

Quality requirements sometimes result in massive documents, up to 500 pages long, especially in:

- comprehensive audits of numerous functions of programmes and organisations,
- the 1990s when rejuvenated democracy rapidly created numerous government programmes that substituted an old centralized command-and-control administrative system (e.g. provision of public services, tax collection, social benefits and privatisation).

Next, a *Management letter* is produced shortly after working papers have been signed by both sides. It usually includes:

- summary of proofs
- reasoning
- conclusions and recommendations.

Complex audits make NIK's management letters complex documents where audit findings and related recommendations need to be traced to original documentation unambiguously. This can be difficult,

taking into account the extensive scope of the NIK's performance audits and the often very intricate nature of problems described.

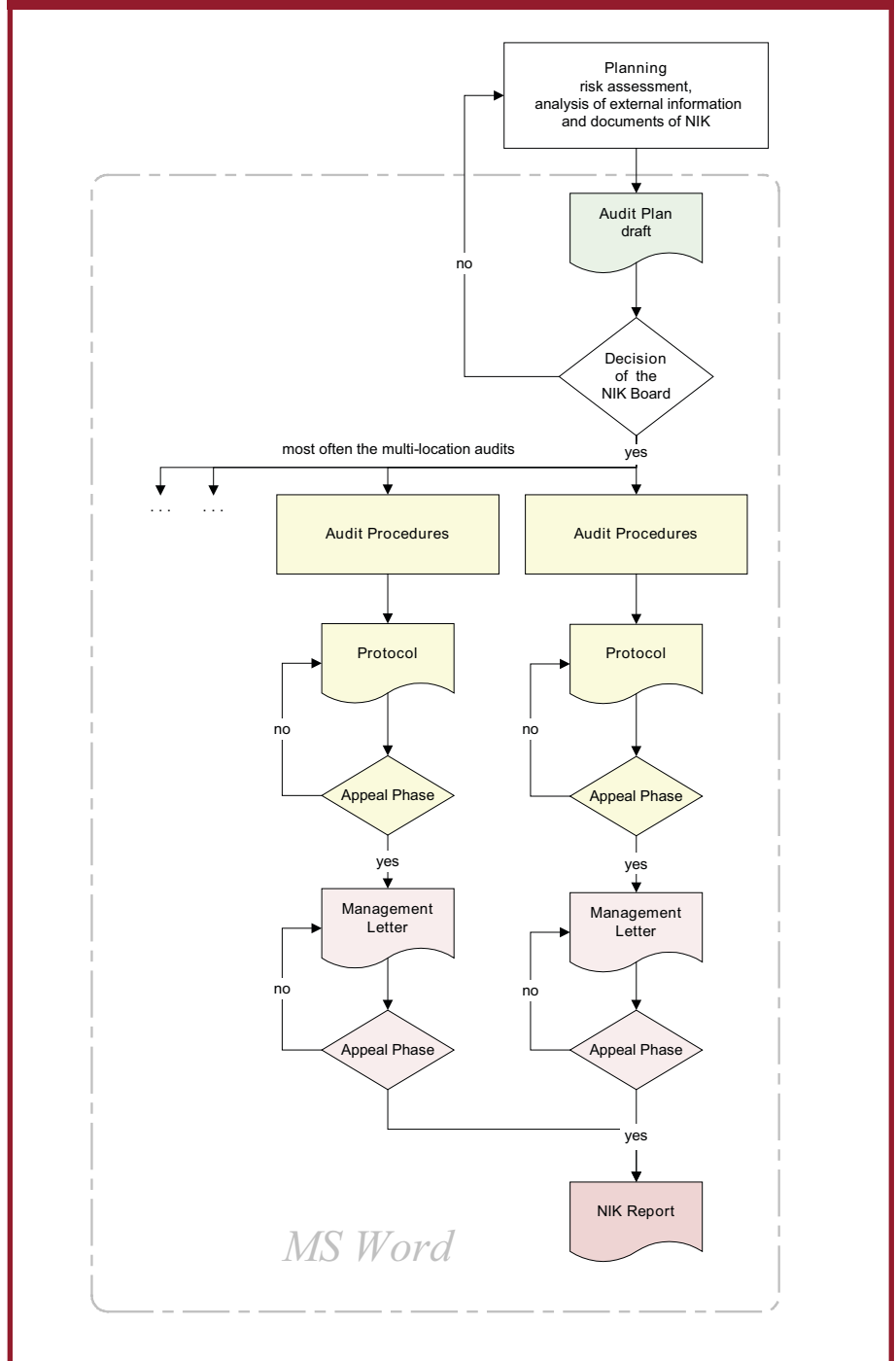
Finally, a public Report is issued. This includes numerous details and references to sources and can easily be more than one hundred pages long. Because of the huge number of documents remaining in the background of the Report, especially when it concludes a multi-location audit, effective audit data management that maintains quality in cross-referencing to source materials can be a cumbersome task. This problem is universal for all SAIs but especially important for the ones that are very active in the performance or VFM fields, where much of important data collected is qualitative and multidimensional.

In this situation there is a natural tendency to search for remedies. It is also obvious that nowadays one should examine IT tools looking for possible solutions.

Computerisation

In the mid-90's NIK underwent rapid computerisation. Our first complex Management Information System was introduced in 1996-1997 and this paved the way for widespread use of PCs and digital formats in NIK's work. All milestone documents (i.e. Audit Plan, Protocol, Management Letter and Report), which were originally handwritten, then typewritten, are now captured in MS Office.

Figure 1 General Scheme of the NIK Audit Procedure – documents produced and archived in MSWord format



Financial CAATs

Since late 1990's NIK has been using CAATs (Computer Assisted Audit Tools) for financial audits during the annual national accounts certification. Since 2001 all government accounts have been audited using internally developed Auditor Assistant software, written in Microsoft Visual Basic. This allows storage, sorting, sampling, basic arithmetic and statistics on a transactional database. Of course, other basic computing tools are used, among them MS Excel and ACL.

More needed

At the beginning of the new millennium we concluded the most extensively IT tool used by our auditors was MS Word. It is without doubt very powerful, but not tuned to the needs of auditors. In fact, the number and variety of MS Word formatting and editing functions drive the auditor to play the role of a professional brochure-maker – not best use of his skills. Official SAI documents (NIK sends out yearly about 3,500 Management Letters and issues almost 200 public Reports), are not necessarily a problem, rather it is a problem with internal documents as Audit Plans or Protocols. We need to focus the attention of our auditors on the content and not on layout.

However, an SAI often has to take texts generated by its own auditors and redraft them to the corporate standard. This is sometimes difficult to manage because of the formatting power of MS Word. At the same time, the size and number of MS Word files makes it difficult to carry out analyses such as simple full-text search or more complex

scanning for specific audit data. Also it would be a poor strategic choice simply to invest in more powerful hardware just to meet the requirements generated by more 'powerful' and demanding software.

The simplest, instant, solution was to tune MS Word. Auditors have been instructed not to format texts freely but to use only limited list of MS Word Styles – defined by NIK in order to:

- standardize document layouts, and
- easily retrieve analytical data with pre-defined macros.

Taming the word processor in this way is inevitable and will continue. The NIK's experiments with MS Word can be transferred to the competitors of Microsoft products. One of the strategic essentials of the Polish SAI is support for open source software.

Is Word Processor A CAAT?

Let us stop for a moment to consider whether we are going the right way. Can a general purpose word processor be called a CAAT? Generally not, if we look at it as at a tool to fill some documents with ideas found elsewhere.

It is the audit methodology that drives the auditor step by step through the audited area and provides him with more specialized IT tools. But the word processor starts to be 'something-like-a-CAAT' when the audited area is terra incognita forcing the auditor to invent ad hoc methodologies, to shape new attitudes and form innovative opinions. When the way of expression is playing bigger role – as usually in performance

audits – the role of writing tool grows as well. In another words, as far as editing means thinking, the word processor starts to be a tool that can influence the auditing process.

Consequently, we can say that text editing software can 'hardly be a CAAT' in financial audit but can be 'much more a CAAT' in performance or VFM audit. Paradoxically, a powerful application like MS Word, in its latest version, invites the user to think: "I can do almost everything with this tool!" The problem is that the auditor does not need to do everything, he needs the word processor to be more a CAAT than a miracle tool.

Desired Functions and Features

While working in the traditional way with a word processor during audits, we started to think about a new tool. We wanted it to:

- 1) replace an 'empty page' with a flexible electronic form – strictly corresponding with the *Audit Plan* – to be filled by auditor with his findings,
- 2) free the auditor from the duty of text formatting – and unify the form of *Protocols* created during field audits,
- 3) treat audit documents as data collections – make their analysis faster and processing easier and
- 4) free our data repositories from formatting code of a *word processor*.

Thus we have created **jNIK**.

Flexible and Open Source

After a series of exploratory visits to countries where SAIs use Team Mate (Denmark and UK) or their own solutions (Czech Republic and Norway) we concluded that for NIK there is no solution available on the market. Apart from the rather obvious ideas of an audit as a hierarchy (tree) of audited problems and electronic forms replacing an ‘empty page’ – ideas used by all visited SAIs – we found many difficulties matching the construction and details of particular solutions to our procedure. The biggest difficulty was how to substantially reduce the risk that the new tool would reform and not deform our way of work – effective as it is, though greatly different from all SAIs we visited. We needed something much more flexible than an off the shelf product and we did not want to enter the long and expensive process of outsourcing our individual project. Help came from the dynamic and developing world of open source technologies, supported by programming skills of our own team of IT staff and auditors.

The authors also decided to expose the project internally with great caution to prevent any false starts that might cause any disturbances to the auditing process and have detrimental effect on the project’s future. We decided to construct **jNIK** and let it compete internally with more traditional tools used by NIK auditors. This will be the best way to find bugs and weaknesses in the design as well as to let this free software grow to meet the needs of the Polish SAI.

CAAT or jNIK

The fundamental decision was to use XML (eXtensible Markup Language) format and the Java programming language. Both are free, widely used by open source programmers, well documented and accompanied by a

multitude of additional tools. The choice of XML was rather obvious – it is now difficult to find any data-processing system not supporting this format. Java’s advantages are platform independence and many XML solutions, many of which are free of charge and available over the Internet.

Figure 2 Technology of jNIK

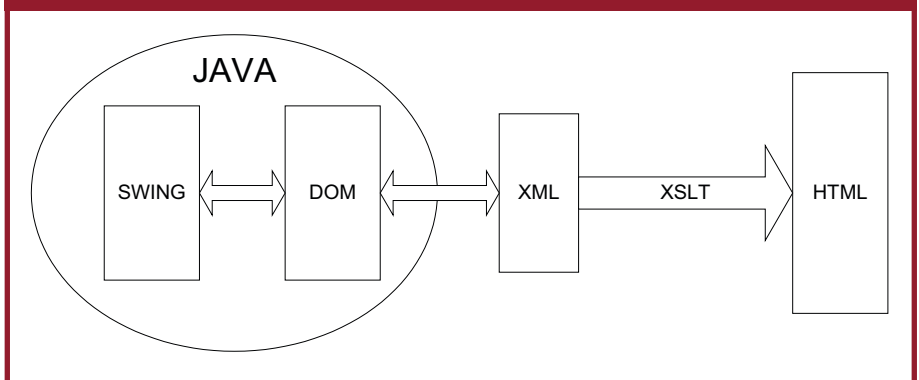


Figure 3 The Audit Plan view

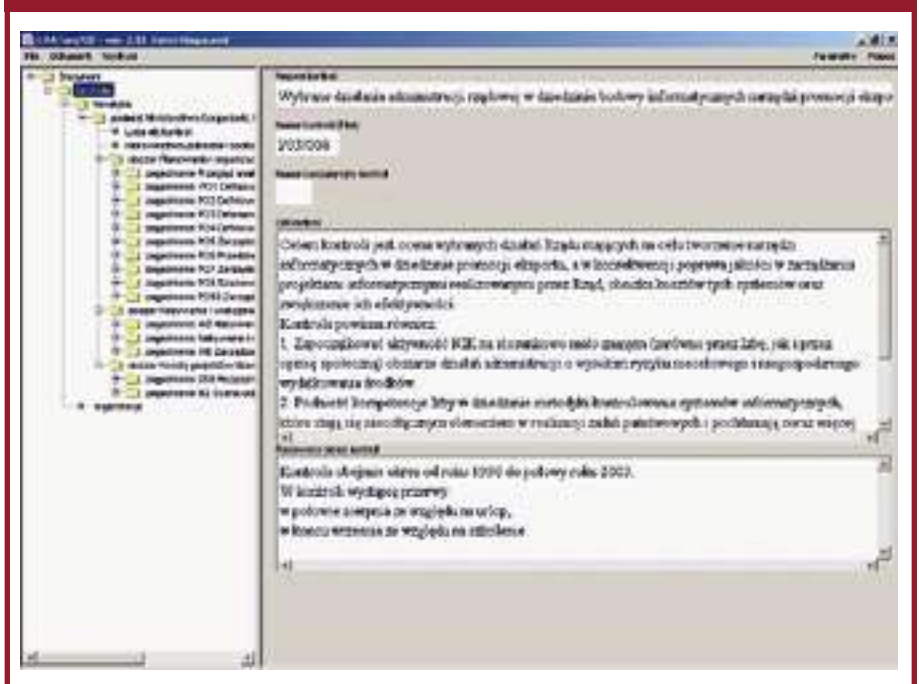


Figure 4 The Protocol view



In Summer 2003, after some weeks of work by a small team, charged simultaneously with other audit and IT service tasks, the first simple version of CAAT or **jNIK** emerged. In the next few months it was used in couple of regular audits.

The user interface is written in Java Swing. The internal XML processing is based on the DOM standard (Document Object Model), being part of JAXP (Java API for XML Processing). Swing supports the essential advantage of XML, i.e. the separation of data from the presentation layer. Basic view is of a tree (javax.swing.JTree class with DOM3 Model) which is natural for XML's well-formed documents. Each branch is connected to an e-form which focuses on the particular part of the XML file. The user can edit the text, though – at least for the time being – the prime role

of **jNIK** is e-form, not text editor. We assume that bigger sections can be easily edited in commercial word processor and pasted into the form.

The same file can be processed from the audit proposal stage, through authorisation by the SAI management, into the full *Audit Plan*, and still be used as the *Protocol* by the audit teams.

The use of the **jNIK** form, naturally disciplines the teams to deal with all points in the *Audit Plan*. They not only have to address every audit area but also to qualify findings or exceptions in an ordered, though not troublesome way. It also gives them an easy way to develop their own findings, extending the *Audit Plan*. For the audit supervisor, **jNIK** allows much easier screening of the team's work, which for multi-location audits can sometimes be a real problem.

In its present shape, **jNIK** helps to organize audit data and stores it as XML documents. The user can initiate the XSLT transformations which, based on the pre-defined templates, can generate the required sets of data or prints. The basic transformation is, of course, XML to HTML, just to view it and print it for signature purposes. Various parsers needed for that operation are readily available. The XML file can be also viewed in PDF format, thanks to e.g. Apache's FOP (Formatting Objects Processor) – giving much more precise result than HTML (although this is not entirely stable yet).

Hierarchy of Document

The XML format does not constrain the types of fields and number of 'audit tree' levels. Nevertheless, when thinking about one Java interface working with XML files on many machines, we found it necessary to make a dictionary of XML elements and limit the number of levels (we used XML Schema as the most precise and flexible tool to do it). An author of an *Audit Plan* has to operate a three level hierarchy:

- **auditee(s)** – pretty obvious;
- **areas** - general definition of a performance or activity areas to be audited;
- **subjects** – as precise as possible definition of sub-areas or matters to be examined by auditors.

Figure 5 Printout of Protocol



Auditors have to develop the 'problem' branches into next two levels:

- **groups of findings** – clusters of audit descriptions with their own commentaries if necessary
- **findings** – basic units of text containing data and short description related to audit findings.

The last level is extended if the auditor happens to discover an inaccuracy, error or mistake of the auditee. In this case the auditor has to describe the causes and impacts, identify responsible persons and evaluate the scale of the problem. He documents points in the Protocol and delivers draft findings that will be used, on reflection and after comparison with other Protocols, in the Management Letter.

jNIK gives the user a set of XML data, containing the *Audit Plan*, the *Protocol* and elements needed to construct the *Management Letter* and the *Report*. It occupies much less disc space than MS Word files containing the same text and, thanks to the XSLT technology, gives practically limitless possibilities for printing formal documents and analysing *ad hoc* generated data sheets etc. XML files will be stored in the file system or database for further automated analysis and processing.

Results

The software is light, currently, the Java source 'weighs' 177 kB, and compiled, 350 kB. But the biggest impact is that **jNIK**, applied to SCC audit procedures, modifies working practices on crucial documents and changes the map of basic software use.

Conclusions

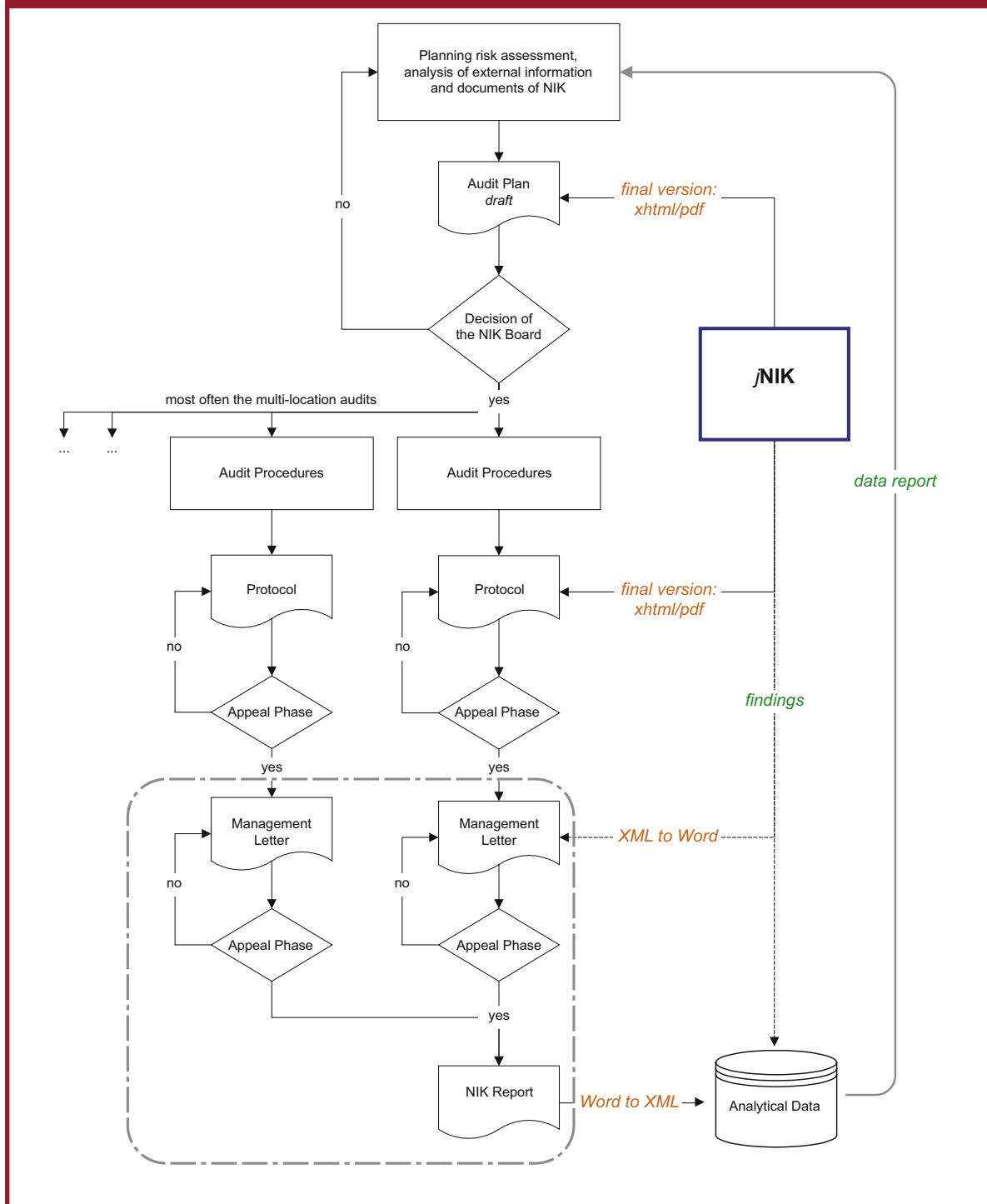
As we experienced, a SAI can easily develop its own software using open source and freeware tools. Simple, flexible and free technology used in **jNIK** (Java & XML) has helped to customize software to NIK's needs. It appears likely that it can be used for the needs of other SAI as well. The benefits of this approach are:

- It fits into the real needs of the organisation, which in case of a State Auditor are often much wider than that of a commercial auditor.
- We have it virtually cost-free. This may help to convince the management that we really need it.
- We have built an in-house support team, willing and able to fill gaps between the auditors and IT people.
- The project supports an interesting and dynamic open source environment, being a counter-weight to the commercial IT market where the monopolistic tendencies of suppliers has grown in the past decade.
- We can spend more money on IT training than buying and supporting expensive software.

Future plans for the **jNIK** include:

- To reach a fully stable 1.0 version,
- Internationalization of the Java code,
- To make **jNIK** publicly available under the GPL licence.

Figure 6 General Scheme of the NIK Audit Procedure – use of jNIK and MSWord



Pawel Banas

(Pawel_Banas@nik.gov.pl)

is a Director for Economy, Treasury and Privatisation in the Supreme Chamber of Control (SCC) in Poland. Despite his numerous management tasks as the head of Department he generates and actively supports IT projects of end-user perspective. Pawel holds a master degree in Oriental Studies and has graduated from the National School of Public Administration in Warsaw. He has 8 years of auditing experience in government auditing, firstly as audit manager, then as Director. He was also heading the General Inspectorate of Customs where he created structures and tools aimed at combating customs fraud. In the SCC he participates in numerous IT projects including work-flow system development, Electronic Documents Storage and Retrieval System and others.



Piotr Madziar

(Piotr_Madziar@nik.gov.pl)

an audit manager in SCC's Department of Economy, Treasury and Privatisation, is involved in different audits of government IT. Piotr holds a master degree of the Warsaw Institute of Technology, Department of Electronics and has graduated from the National School of Public Administration. He has 8 years of auditing experience. He was also Director General of the Polish Agency for Foreign Investments where he was involved in various projects of technological nature. In the SCC he helped in creating and implementing internal standards for IT auditing. Piotr was also involved in IT audits using COBIT framework criteria for the evaluation of good practice in government IT.



Wieslaw Sawicki

(Wieslaw_Sawicki@nik.gov.pl)

an auditor in SCC's Department of Economy, Treasury and Privatisation where he is involved in audits of government IT. Wieslaw holds a master degree of the Warsaw University, Faculty of Mathematics. Recently he completed an IT security program of the Warsaw Institute of Technology. He has over 20 years experience as the programmer and project manager in IT systems development projects (banking) and 2 years experience in government auditing. In the SCC Wieslaw participated in IT audits in the area of Customs, export promotion and others.

