

UDK (UDC): 930.25:004.42

Tipologija: 1.09 Objavljeni strokovni prispevek na konferenci
Category: 1.09 Published Professional Conference Contribution

*Martin Stürzlinger**

STANDARD SOFTWARE FOR AN ARCHIVAL INFORMATION SYSTEM: BETTER, FASTER, CHEAPER

Izvleček:

Standardna programska oprema za arhivski informacijski sistem – bolje, hitreje, ceneje
V članku avtor obravnava uporabo različne programske opreme za arhivski informacijski sistem. Predstavlja prednosti in slabosti individualne in standardne programske opreme, ki bi jih naj arhivi upoštevali, ko se odločajo o licenciranju novega arhivskega informacijskega sistema. Nekateri arhivi ponavadi razmišljajo o zamenjavi individualne programske opreme ali standardnega produkta, ko ta več ne zadostuje njihovim potrebam. Drugi še uporabljajo iskalne pripomočke na papirju ali datoteke, kot so MS-Excel ali MS-Word. Razvoj vsake programske opreme prinaša neko tveganje. Zelo pogosto so izračuni za individualno programsko opremo prenizki. Za to opremo lahko stroške le ocenimo in tvegamo, da bodo 20–25% prenizki. Za večino standardne programske opreme velja, da so v enotno ceno všteti vsi stroški in ne predstavlja nobenega tveganja.

Ključne besede:

arhivski informacijski sistem, individualna programska oprema, standardna programska oprema, razvoj, povratne informacije strank.

Abstract:

Standard Software for an Archival Information System: Better, Faster, Cheaper
In the present article the author discusses the use of different software for an Archival Information System. He presents advantages and disadvantages of individual and standard software, which archives should take into account when considering licensing a new Archival Information System. Either they are changing from their individual software or from an older standard product that does not fulfill the requirements any more. Others again use either paper based finding aids or files like MS-Word or MS-Excel. Each software development carries a certain amount of risk. Very often the budget for individual software is estimated too low. For individual software one can only estimate these numbers and runs a high risk that these numbers are 20-50% too low. For most standard software products one gets all of these cost factors for a fixed price - and at no risk.

Key words:

Archival Information System, individual software, standard software, development, customer feedback.

INTRODUCTION

In the „early days“ of personal computers the question of an Archival Information System (AIS) was quite simple to answer. There was only the “do-it-yourself” choice. If an archives wanted to support its activities with software it had to do everything itself - from design, planning, financing, programming, testing and training. At a time when libraries would have considered such an approach as absurd, many archives fought for

* Dr. phil. Martin Stürzlinger, MAS, MLS, Scope Partner, Albertgasse 33/17, A-1080 Wien, Österreich.

themselves. But then again, how many archivists thought or even knew about archival standards.

Usually an archivist with programming knowledge or a very willing IT-department who would be interested in such an unusual task, were required. While some archives started to use these in-house developments most others continued with a traditional paper or basic MS-Word and MS-Excel approach for describing their fonds. In this way archives invested substantial personal or financial resources in the development of software. They were planned, developed and used in-house. No other usage, no other users were envisioned. This also continued an archival tradition of not collaborating and not sharing resources. Because each and every archival holding and its description is unique also the task of describing it was considered unique.

WHAT IS STANDARD SOFTWARE?

Up until the mid-1990s there was also not much choice to an in-house development as there was hardly any software around that could claim to be “standard”. But what is the “standard” in standard software? It is not software to a certain international or national “standard”. Standard software does not necessarily adhere to archival standards like ISAD(G), ISAAR(CPF) or others. One has to differentiate between the different meanings of the word. The word “standard” reflects to the fact that it is a “standard product” like a car or a product in the supermarket. It is offered “as is” to everybody on the market: “What you see is what you get”. The customer has to decide about the qualities and the functions before buying. Many software programs are sold like this: Adobe® Acrobat®, MS-Office®, Oracle Database® and many others.

Actually one does not “buy” software because this would mean buying the source code of the software. In the case of Adobe® Acrobat® this would probably come close to buying the whole company - something only very few people are interested in. The right term is to license software - more precisely to license the right to use software, in most cases for an unlimited time.

When licensing standard software, the customer gets the software as seen and tested. Changes in the appearance and features are generally not possible. This is comparable to buying a car where one can choose a brand and then a model within the brand. Each model has a list of extras one can choose from. Generally it is not possible to choose for model B an extra that is available only for model A. In another example, when using MS-Word one has to be satisfied with the features and the appearance provided. It is not possible to have the close button in the top left corner - it will always be in the top right corner. Of course standard products can be customized - adapted to individual needs or wishes. One can add a feature to a car that is not in the extra list or program a plug-in for MS-Word. But this usually comes at a higher price. It is similar with standard software. Some come as a single system at one price, some other have different “models” each with a list of “extras”. If a system has this flexibility it can adapt to different needs of the customers and also be flexible in price. If a big archives has lots of requirements it will use the “big model” with lots of “extras” and if a small archives only has basic needs it will stay with a “basic model” with no “extras”.

Another big difference between individual and standard software must be mentioned. Individual software is created according to a requirements specification and promises to work exactly the way the customer has defined it. This software producer guarantees all the functions specified, depending on the contract. If an error occurs he is liable to fix it.

This is different with standard software. Here the producer does not guarantee that the software works flawlessly. This sounds surprising but is something we routinely accept with many products. Looking at the homepages of the above mentioned software companies it is easy to find a long list of known errors in the software. Only very critical errors that endanger the overall system or the saved data need to be fixed right away. If you happen to find such an error for example in MS-Access and report it to Microsoft they will react to this and fix it with a software patch as soon as possible.

As any good vendor, software companies are happy to receive customer feedback - especially on errors in the software. These errors are then evaluated to see how critical they are to the customers. According to this judgment a decision is taken on when to fix it. If it is more severe it will probably be fixed with a software patch (which repairs just the function that is not working). If it is only minor the repair will probably be delayed until the next version of the software is released. This version then includes everything from new features, better working features and bug fixes.

WHAT IS AN ARCHIVAL INFORMATION SYSTEM?

In this article I concentrate on the term Archival Information System (AIS). What does it mean and why is it used? An AIS is a software for the whole archives. Its main focus is on the archival holdings, concentrating on the description of fonds, series, dossiers and documents in combination with thesauri/descriptors. But there are many other tasks in an archives, which should be solved in an integrated way. Most important of all a modern system should be able to deal with paper based and electronic records in an integrated way. For the descriptions there should be one unified way to administer them; there should be one way to search for them. No switching between systems or dealing with two logical structures should be involved. An AIS (or sometimes called AMS - Archive Management System) must also be capable of supporting the total of work processes in an archives. This ranges from pre-archival work, through accession and description to loan and publication on the Internet. Description and printing finding aids are the central part - but only a part of the duties of an archives. The central software for an archives should cover most or all of the aspects that the institution deals with.

When software is selected, a great deal of importance is attached to sophisticated individual functions. Later, in operation, the sum of the functions and the uniform mode of operation turn out to be more crucial. It is important that one system supports all common work steps within the archives and in contact with the outside world (accessions, loans, data import and export). Independent databases for image or literature collections are not necessary, paper and electronic files should be managed in a single system. This uniformity creates the crucial added value, management and searching is made considerably easier.

ADVANTAGES OF STANDARD SOFTWARE

Standard software has many advantages for an archives over individual software. The advantages can be grouped into 3 categories:

- Number and quality of functions
- Reduction of risk
- Time

NUMBER AND QUALITY OF FUNCTIONS

Standard software usually has more functions than individual software. The reason for this is simple. It is programmed and developed by a company that cares about its product and its customers. To have a viable chance in the market selling the product, it has to offer a wide range of functions to satisfy all the diverse customer needs. The company developing the software has different customers with individual needs. There are big archives, small archives, company archives, special archives etc. When offering standard software the product has to be attractive to all of them. In order to achieve this it is necessary to include a wide variety of functions - many more than would be included in individual software that focuses on one customer only. Usually these functions will also work better than in individual software. One of the strategic advantages of such a software producer is a broad customer base. If the producer listens to its customers and understands the needs it will try to accommodate them as much as possible. Therefore functions are often developed in close contact with the practical needs of many archivists doing the work. In individual conversation, in workshops or in other forms of feedback the producer can shape the software exactly to solve real problems. With this knowledge, functions can be refined in a better way considering different aspects of solving the same problem.

Individual software very often starts from one individual point of need for which it is developed. Once the system is working it grows and grows in functions and is adapted to other areas. One example for this is when a library or museum software is taken and adapted to the needs of an archives. Another example would be an image database that is adapted for the description of modern administrative records according to ISAD(G). While such a system can grow to be useable the limitations of the original design will always be visible. Standard software - at least good standard software - has taken most needs of an archives into account at the time of the original design. An integrated workflow, a uniform interface and a uniform way of handling things bring an ease of use and efficiency. Important features like flexible data import and export as well as reports will be available.

The large customer base has another important advantage that reflects back on all the customers. Standard software will always include solutions that were not regarded as mandatory when the software was licensed. But when using the software one learns to see the advantages of these new functions. If you take a very bright archivist who is responsible for an individual software project you can imagine that he thinks of 50 very clever functions for his software. Due to the larger feedback and more resources the standard software will easily include another 100 functions that this person has not thought about - but will make his life easier nevertheless.

Also this clever archivist has to constantly be up to date and change the software to the changing needs. Not a year can go by without some minor reworks, every 5-7 years the software has to undergo a major overhaul. All of this time and energy is lost for the core tasks of an archives. A user of standard software on the other hand can more easily focus on these tasks and relax on the software side. Software with many customers will always be up to date and it will often include solutions that are ahead of time. A large and diverse customer base should also be a reassuring sign for flexible and adaptable software. If such an AIS can be used in a national archives, a company archives, a clerical archives and a university archives in different countries chances are high it will be suitable for many other archives, too. It then has a data model that can adapt to their different requirements and archival traditions.

While individual software can - provided enough resources - have many features and fulfill many needs, there are some challenges too steep for it. Quite often it is the size or the complexity of the database that makes it impossible to further develop it. More and more resources have to be invested to keep the system alive. And some challenges should not even be tried with individual software, like the complexity of electronic archives. All the complexities surrounding the OAIS model for preservation of electronic records should be either solved with huge amounts of money or standard software - a combination of both is best of course.

Standardization leads to a focus on strategic processes - away from perfecting rare special cases. Standard software suggests the use of tried and tested sample processes for many workflows and usually supports description standards like ISAD(G), ISAAR(CPF) as well as EAD and EAC without prescribing them to the archives. The co-existence of standards and historically developed structures should be part of a good software.

REDUCTION OF RISK

If an archives is looking for software to solve its needs it should turn to standard software not only because of better functions but also to avoid the high risks involved in the development of individual software.

When developing individual software most institutions only see the price of the programming. They estimate how long the project will take, multiply it with the hourly rate of the programmer(s) and have calculated the budget. Rarely other factors are taken into consideration. In the specification phase - when all the functions are planned and designed - it needs the expertise of the archivists and the programmers combined. During this time, which can easily be up to two years, archivists cannot fully work for their real job. In an honest project the time of programmers and archivists have to be calculated toward the budget.

During and after programming, each function has to be tested and - if the test fails - corrected. Then documentation for the whole software should be written. Without it one is always dependent on the two, three persons involved in design and programming. After this, training and the time for yearly maintenance have to be considered. All of these factors must be calculated for a valid budget. For individual software one can only estimate these numbers and runs a high risk that these numbers are 20-50% too low. For most standard software products one gets all of these cost factors for a fixed price - and at no risk. Because one knows that standard software is existing and working - with individual software there is always a risk that the project fails.

The biggest risks in any project are the changes to it. This is especially true for software projects. I have seen many projects for individual software that started without a clear specification listing all the functions needed. Programming started with a set of functions but very soon changes, additions and other priorities were added. Without a clear project plan, a budget and most of all a deadline such a project is at high risk of failure. If one manages this phase eventually the archivist will start to work with the core functions that are ready for use. But the overall system will always be incomplete and unfinished. More resources, many more meetings and more budgets are needed - often resulting in frustration.

Individual software is always dependent on the resources of the institution that commissioned it. If it runs out of money, manpower or energy to further support it, the

software is already outdated. Two to five years later it becomes unusable and the investment is lost. Standard software on the other hand is dependent on a much broader customer base. This guarantees that the software will be stable and up to date. The original investment at the beginning is preserved or even enhanced.

One of the strategic advantages of standard software is the regular product maintenance. The producer of an AIS has a high interest in keeping the software attractive for new customers and will fix errors, make features better working and add new features. All of this usually falls under regular maintenance which is essential for preserving the value of the investment the archives has made. Product maintenance comprises the regular delivery of patches, releases and versions including documentation and installation instructions. It consists of:

- Value Preservation
- Correction of program errors and minor defects within the framework of systematic release management
- Adjustment to a developing technical environment (operating system, database).
- Immediate recovery of serious faults or operation-critical errors (patches)
- Value increase by incorporation of new functions and technical possibilities

As a rule of thumb around 1/3 of the money invested into maintenance is value preserving and 2/3 is value increasing. Customers have a significant influence on the selection of value increasing investment. You can make suggestions and report faults and wishes via the feedback system. Therefore regular maintenance not only keeps your software working but it will usually also make it a better product.

If the producer of standard software wishes to retain its position and remain attractive, it has to constantly keep the software up to date. Due to the customer base archive-related or technical enhancements can generally be developed faster and at a cheaper price because many institutions (customers) share the investment.

Future requirements are covered before the majority of archives actually feel a need for them - the availability of new functions even prompts many archives to start thinking about them. Standard software can therefore be also the bearer of new technologies and ways of thinking. This is true for example with EAD (Encoded Archival Description). While most archivists know about it only few archives have developed a real need to use it. Chances are high that a standard AIS will provide an EAD-Export before the archives sees a practical need for it. Once the AIS provides the function this can become the starting point for a project to share descriptions and present them in the EAD format.

If the standard software has a modular structure it enables stepwise implementation - geared to concrete requirements and without gaps. This may start with a module for description, then for accessions and continue to the Reading Room. As the latest stage a module for the Internet may be added resulting in a wider range of users. By adapting to the growing requirements and financial possibilities the archives only has to invest what is really needed, when it is really needed.

In some countries archives or their IT-departments argue that it is cheaper to develop individual software because the hourly rates of the programmers are substantially less than in other countries - 30%, 50% or even more. This fact cannot be negated but it only concerns one factor of the price. There are many other factors that determine the overall cost of software, for example:

- The programmers of the standard software are already experienced in the topic not starting from scratch and should therefore be more efficient.
- The software vendor of standard software does not sell the product purely based on hourly rates. The license price will always be much lower than the costs of all the hours needed to create the software as the vendor can sell the product multiple times.

All the arguments in the chapters about the functions and time - and most of them in this chapter about the risk - are still valid. Therefore standard software that already has a stable customer base can offer more functions, work better and still be cheaper overall than individual software - no matter the hourly rate of programmers.

TIME

Creating software takes considerable time and effort - especially for an institution like an archives, which has no trained experts for this task. First one needs to find programmers. Then the archivists have to write a list of requirements which is translated for the programmers into a specification. Then the main task of programming starts which is followed by a testing phase. The process is concluded by documenting and training the software.

All of this takes a lot of time and cannot be achieved within a few months - if the software is considered to have more than just the most basic functions. Selecting and introducing standard software can be done within weeks. If an archives has to go through a public tender the effort is considerably higher because the requirements have to be listed in general and a big administrative overhead has to be dealt with. But even with these 6-12 months needed for the tender process standard software should always be faster ready-to-use than individual software.

When creating individual software each project step needs to be done or accompanied by at least 2-3 archivists. During the development phase of the software these persons can only work in a much reduced manner for the other tasks of an archives. The consequence of this is that the software does not increase the efficiency of the archives but for a considerable time decreases it. More time is lost in the long run when constantly maintaining the software. At least in a yearly cycle the software has to be reworked and every 5-7 years it needs to be reworked in large parts. All of this needs lots of time, energy and money which are lost for the main task of the archives.

DISADVANTAGES OF STANDARD SOFTWARE

The list of disadvantages of standard software is clearly shorter than the list of advantages. Or differently phrased: individual software seems to have some advantages.

When designing functions for individual software one usually tries to implement just those needed at a given time. Elements not needed or that might be interesting in the future are disregarded. This can make the software in its initial phase easier to use. It works the way and uses the terms that people are used to in this archives.

These advantages are short lived and short sighted. Once people start to use the software they will immediately grow an "appetite" for more and better functions. The result is often a whole range of functions that grew without a "master plan". Good

standard software starts with such a “master plan” and an accepted terminology. This makes it easy to grow in functions and easy to understand for the individual archivist.

Individual software can be seen as more flexible as it includes just the flexibility needed at the time of creation. But very soon other areas will be identified where flexibility is needed. Also when comparing individual to standard software often one or two areas are cited where the individual has a more flexible or better solution - ignoring 50 or more parts where it is the other way around.

According to my experience the most important argument for individual software is that other interests can be accommodated - other than trying to find the most efficient, best, cheapest and fastest solution for an IT-problem of the archives.

Quite often archivists have a personal interest in creating or keeping “their” software, to keep control of “their” project. An AIS is a central instrument for many archival tasks. Controlling the software for these tasks gives the person an important position. Losing the individual software can lead to a decline in importance.

IT-departments usually have people with the job and the interest to design and develop software. Very often they will find arguments that support the development of individual software for two reasons. Firstly, because this is how they have been trained and how they view software. Secondly, because this will increase the importance of their person and their department. As an AIS is a software project their arguments are often considered more valid than those of archivists - even by archivists themselves.

With the political or administrative leaders of archives a whole slew of other arguments are introduced into the discussion. Many individual, personal or region specific arguments are used in this area. One typical argument in this category is to defend “our own” against a “foreign” development. Another classic is the argument of making a difference between the capital and the center of a region/country against the “provinces”. One would be surprised to see how often a regional archives chooses a certain product because it does not want to use the same as in other archives. Very rarely do logical reasons or arguments about functions, risk or time play a role in this area.

CONCLUSION

Today many archives consider licensing a new AIS. Either they are changing from their individual software or from an older standard product that does not fulfill the requirements any more. Using standard software enables an archives to respond correctly to challenges. A wealth of experience and dozens of man years of knowledge acquired in the development and support of a large number of customers result in an off-the-shelf software package which covers nearly all individual requirements. At the same time, it guarantees that the software will also remain up-to-date in the future.

The use of a standard software package is a strategic decision. The customers’ confidence in the product is ultimately the most convincing proof of this. The benefit derived by the archives from using standard software depends less on individual software functions than on their entirety and on qualitative criteria. On the whole, the work of archivists is rendered more interesting and more attractive and avoids running a software development shop.