Abstract: Since the 1970s, computer supported data processing has been implemented in the laboratory and laboratory information systems (LIS) are being developed. In the following years, the programs were expanded and new laboratory requirements were inserted to the LIS. In the last few years, the requirements have grown more and more. The current tasks of the LIS are not only the management of laboratory requirements but also management of processes, data security and data transfer and they have become very important. Therefore, the current monolithic architecture of LIS has reached its limits. New methodologies like service oriented architecture, e.g. microservices, should be implemented. Thereby different specialized manufacturers provide software for one or a few tasks. These tasks can be more easily actualized like in the new field of agile software development. This new concept has been designed to provide updates and customer requirements according to its new organization structure in program development in a short time. For efficient data transfer, new interfaces and a standardization of master data like logical observation identifier names and codes (LOINC®) are advisable. With the growing data transfer, data security plays an increasingly important role. New concepts like blockchain programming (e.g. Medrec) are currently tested in (laboratory) medicine. To get an overview of the requirements of the own LIS, an Ishikawa diagram should be created. The main points of an Ishikawa diagram are shown and discussed. Based on the today-collected data, expert systems will be developed. For this kind of data mining, a structured data exchange is necessary.

Keywords: blockchain; ethereum; laboratory information system (LIS); LOINC; microservices; scrum.

Introduction

Normally, in medical laboratories, only the analytical proficiency is observed and appreciated. Media and television support this view. The laboratory information system (LIS), a guarantee for data management, is noted only in rare cases and operates silently in the background.

This article gives an update and is an extension to our work in 2014 in which we provided an overview of the requirements for a powerful system for laboratory data management [1]. Since then, technologies have changed and requirements for laboratories are still growing.

LISs were introduced into clinical chemistry in the 1970s [2] and have become an integral part of laboratory life. It is the basis for proper sample acquisition, supply of analytical results from the measurement systems, medical reporting and billing. In addition, it is responsible for any external test facilities and archiving of the samples. Although continual improvements are being made, there is a growing number of people (laboratory personnel, clients) calling for a better, more powerful and contemporary LIS. Ideally, it should be possible to recall the status of a sample like the status of a parcel from a parcel service.

It has to be kept in mind that in the near future we will have an increasing amount of data (cf. big data) especially from the “Omic”-sciences (metabolomics, pharmacogenomics, etc.), individualized medicine, gene sequencing, microarray testing and of newer techniques such as nuclear magnetic resonance (NMR), next generation sequencing (NGS) as well as matrix-assisted laser desorption/ionization time-of-flight (MALDI-TOF) spectroscopy. A new generation of measurement systems
is expected to become part of routine diagnostic procedures including nanodiagnostic systems (quantum dots [QDs], gold nanoparticles, cantilevers) [3]. According to the demographic trend in some countries, the number of tests will increase. The increasing number of analytes (e.g. in drug monitoring) and type of sample (e.g. saliva diagnostics) also expand the amount of data to be processed.

Problems can arise in the sample receiving and sample distribution due to the increasing number of analytes and measurement systems. Additionally, data reporting should be restructured according to new techniques and measurement values produced by, for example, microarray testing or gene sequencing. The additional data must be presented in an adequate manner to the client. Furthermore, a better harmonization between different laboratories is necessary [4]. The compatibility between the computer systems saves time and is beneficial to the patients [5].

Some questions arise such as: can the contemporary LIS manage new tasks? Is it sufficient to add parts to the contemporary LIS or should the LIS be completely restructured?

Overview of the tasks of a LIS

There are four main pillars concerning the technical needs of a laboratory (Figure 1). Each of them is in direct or indirect contact with the main technical system of any laboratory – the LIS.

The main pillars are:
1. Management of processes
2. Management of data security
3. Management of data transfer
4. Management of laboratory requirements

Management of processes

Management of processes means that all processes and workflow concerning the laboratory should be monitored and evaluated for improvements. In this context, the keyword “big data” is often used. In laboratory terms, it means sample tracking and process monitoring.

Actually, the LIS and other systems that interoperate with it store and collect different but potentially usable data that are important in a laboratory. One interesting point in each laboratory is the performance of a laboratory compared with another one. Actually, it is hard to exactly analyze why some laboratories work more efficiently than others. Here, the usage of big data algorithms may help to optimize laboratory process and compare workflow steps of different laboratories [6, 7]. The best way would be to track the whole process end to end. The process starts with the request of the customer and then collection of samples by the dispatcher or taxi driver to transport them to the laboratory. In the laboratory, all samples and only corresponding orders must be brought together. The idea is to track each sample to prevent from confusion of patients. In the laboratory, all necessary work is done and a report is sent to the customer. The last process step in the value chain is billing.

For the management of processes, laboratory workflow profiles are very useful. Some organizations like the IHE (Integration the Healthcare Enterprise), a group of healthcare professionals and partners from the industry, provide very useful workflow profiles (see http://www.ihe.net/Laboratory/).

Management of data security and data protection

In May 2018, a new law concerning data protection was introduced in Europe. The main goal is to harmonize data privacy laws in all states of the European Union (EU). The new law is called “general data protection regulation” (GDPR). Due to potential enormous fines for violations, companies have to adapt to the new situation. The
new regulations will be stricter. Some countries have not yet been aware of any data protection or to the same extent. Here, many LIS will have problems to fulfill all legal requirements, for example, the so-called “right to be forgotten”. This right means in short, that all personal data have to be erased when the (personal) data are no longer necessary for the original purposes. This requirement cannot easily be handled for current LIS and is also a problem for those companies using laboratory data management systems (LDMS). Here it is necessary that the LDMS conforms to the GDPR [8, 9].

Management of data transfer

The actual standard formats used for data transfer like LDT (labor data transfer) and Health Level (HL)-7 are known in the healthcare market and have gained successors (LDT 3.0 and Fast Healthcare Interoperable Resources [FHIR]).

In the old standard format LDT 2.x, it was not possible to add the report of the laboratory in the form of a document, for example, as pdf. It was only possible to transfer the report information in text format and so it was impossible to transfer graphics like electrophoretic titration curves. As a result, customers often received paper reports in addition to the transferred digital report. The new standard LDT 3.0 now offers the possibility to embed documents. So, reports including graphics can be attached as rendered by the laboratory software systems. This is a great benefit because paper consumption is reduced and customers can automatically include the digitally received report to their software systems. For legal reasons, it is necessary to use a digital signature and end-to-end encryption to transport the digital report. Therefore, an appropriate infrastructure must be provided and the workflow for report delivery must be adapted. However, there is still a lack in the exchange of medical data between different systems and companies [10, 11]. The main challenge is still in mapping differently used names for the analytes and checking if the method and reference range are identical.

Management of laboratory requirements

The main tasks of a LIS are shown in a fish bone diagram (Ishikawa Diagram; Figure 2). We revised our diagram as presented in our previous article and added a few aspects
that will become more important [1]. It is clear that only the main tasks of a LIS can be shown in such a diagram. The most important sub-items (e.g., quality management) are described as tasks.

In the following paragraphs, useful and necessary requirements are described. The efficient realization cannot be done for a monolithic system, but for a system of microservices. Most mistakes occur in the pre-analytical phase and not in the analytical phase, as many people believe [12, 13]. The pre-analytical phase is very vulnerable to uncertainties [13, 14]. Around 62% of the “laboratory” mistakes are made during the pre-analytical phase, followed by the post-analytical phase (23%) and the analytical phase (15% [one point for the low amount of mistakes in the analytical phase is the high standardized automation and the LIS support]) [3].

In the scope of the pre-analytical management, all marks on requisition documents should consist of master codes independent from a special LIS. Requisition documents are (in some cases and countries standardized) documents where doctors can write down or mark their orders to a laboratory. Each LIS must not only understand these master codes but can also use its well-known old codes if a mapping table precedes the order import process. A great benefit would be that each laboratory could use the same well-proven requisition documents which would result in further cost and error reduction. These master codes could also be used for electronic order entry between different LIS. This means not only the order import process should use a mapping table but also the order export process. All LIS would only use master codes when communicating with each other. The master codes should be stored in a centralized database which should include pre-analytical requirements/preparation of patient and sample material, method information, measurement frequency and reference values (sex, age or in veterinary medicine animal species). Preferred and alternative sample material should be defined. The data must be regularly updated (automatically because manually updates are often forgotten). Moreover, the recognition of special procedures, for example, health insurance, must be possible. The installation of the master data (e.g., from patients, analytes) and entries must be easy and also available for other programs (e.g., billing, statistics). Here the established standard library (Logical Observation Identifier Names and Codes; LOINC®) could be the medium of choice [15–17].

In the above section, the meaning of master codes is presented. But how can a master code be organized and who creates them?

All sets of master codes should have the same structure (Figure 3). One possibility is to use substructures. For one analyte, the master data should consist of a “fixed” and a “laboratory specific” part. The “fixed” part contains, for example, the general and laboratory independent information about the analyte. The “laboratory specific” part can be further divided into a “general” and a “variable” part. In the “general” section, the reference range, the used method and other information of the analyte should be inserted. The “variable” part can be split into an “external relevance” and an “internal relevance” part. In the section of “external relevance”, the used material or

![Figure 3: Possible structure of a data set of master codes: the master code can be built up by a “fixed” and a “laboratory specific” part. The “fixed” part contains general information about the analyte. The “laboratory specific” part consists of a “general” as well as a “variable” part. The last part can be divided into “external” and “internal” relevant information.](image-url)
comments to the used method can be inserted. The “internal relevance” part can contain information about billing, shipment or distribution in the laboratory. The main advantage of this structure is the easier exchange of data because export relevant parts can be defined (Figure 3: “for export to other programs”) as well as a country-specific part, e.g. billing modalities or a laboratory-specific part, e.g. distribution of the samples is taken into account (Figure 3: “internal use”). Because of the general structure it is also possible to use it for microbiology, transfusion medicine and further developments or further routine analytics like genetic sequencing.

In chemistry, the International Union of Pure and Applied Chemistry (IUPAC) is the recognized world authority in developing standards for the naming of the elements and chemical substances. May be it is possible that the International Federation of Clinical Chemistry and Laboratory Medicine (IFCC) will fulfill the same task by the standardization of the abbreviations of the master codes.

The electronic order (order entry) is an easy and safe form of the requisition of laboratory analytes. In the last few years, a lot of innovative software solutions have come to the market. In Germany, since 1 July 2017 it is no more necessary to transfer a standardized piece of paper like the “Muster 10” and “Muster 10A” for the order of laboratory values with all order information from customers to the laboratory for legal reasons. In this case, the whole order can be transferred digitally – only the sample must still be transported manually, but it is necessary to digitally sign the order with a qualified signature. This step is still difficult to handle as well for the customer as for the laboratory – because all concepts were based on retrieving a piece of paper.

The graphical user interface (GUI) of an order entry system should allow the creation of special templates for each entry/customer and also for pre-analytical requirements and patient preparation. For each analyte, the preferred sample materials should be shown when ordering the analyte in the form of an interactive program.

Artificial intelligence (AI) can support the requisition by suggesting analytes in, for example, differential thyroid diagnosis, combination of tumor markers or in the staged diagnosis of vitamin deficiency (vitamin B12/transcobalamin). Self-learning systems can analyze the requisition behavior and alert the user if he or she has forgotten relevant analytes (AI/self-learning systems with user [doctor] interaction are only possible in the case of order entry).

It may be useful that decentralized point-of-care testing (POCT) data are made available for a better interpretation of all analytes measured from the patient and for archiving of the data obtained.

One important item for quality improvement and sample tracking is the registration of external samples and the transport box on collection for verification.

In the section of the pre-analytical phase, the transport box delivery should be electronically documented (it can be combined with the sample registration). This is not only for documentation and sample tracking but also for verification of the delivery of the transport box/sample by the carrier to the laboratory. We developed a simple program to track the check-in and -out of transport boxes and store all relevant information in a relational database. Most important information is which person (at which time) did a check-in or check-out action and it is useful to assign this data to the corresponding tour. After the introduction of the software, the result was a detailed history of tour arrivals and the sum of boxes a person checked out in history that never came back. Each tour had a target time for arrival. By calculating the delta of target arrival time and real arrival time over longer periods, it was possible to optimize the traveling service.

Sample registration can be done manually or better automatically (in the case of order entry) and is more than only an administrative act. It constitutes the first visual inspection (e.g. examination of the sample for physical integrity). For an uncomplicated sample registration or registration of requisition, specialized templates should be used. If scanning systems are used, at least an optical character recognition (OCR) system should be installed. More efficient and fail-safe is the usage of optical mark recognition (OMR) and two-dimensional (2D) barcodes. OMR can be used to capture marked data from document forms, here order forms. Doctors get order forms (paper) with available analyses/tests of the laboratory. To each analyte a markable box is assigned and underlying information to each box is stored within the scanning system. When doctors order analytes from the laboratory, they only have to mark the boxes of the desired analytes instead of writing down the order in clear text. This is clearly more efficient because customers only have to mark the desired analytes and the laboratory knows exactly the required tests and can automate the order registration with the help of scanning systems. Scanning systems can recognize which boxes on the order forms are marked and deliver the underlying information (the short name of the analyte) to the LIS. The 2D barcodes can additionally store all patient and customer information. The scanning system can easily retrieve all this information by reading only the barcode.

By using manually registration of requisition special templates/input screens for clinical chemistry and
microbiology are possible. A further step is that analytes are registered and a special software allots them to the different sections of the laboratory, e.g. microbiology, transfusion medicine. This can be a useful and time-saving tool for smaller laboratories with a basic program of analytes. After registration of the analytes, a software should check the frequency of ordering the analyte (a weekly measurement of glycated hemoglobin [HbA1c] is not necessary) and in a further step the combination of the analytes by using AI. Requisition documents are normally automatically scanned and interpreted. The software often has problems in differentiating between, for example, the characters I, l, 1. A self-learning system (learning from the user correction) and/or the usage of proven datasets (prenames, cities, streets, etc.) would help to reduce this error rate.

In laboratories that also analyze veterinary or pathology samples, it is clear that different input screens have to be used. A further necessity is that a special program part/program module is responsible for the distribution of the sample (e.g. in-house testing, external testing). If different requests from one patient are ordered (microbiology, blood group and clinical chemistry), separate numbering is used for each requisition in many laboratories because of the splitting of the samples for analysis in different sections of the laboratory. Future systems should only use one number for one patient and requisition. This is helpful for the incorporation of the measured data into a complete medical report.

In view of the analytical phase, new program modules must ascertain that data transfer for additional analytical techniques (NMR, nanodiagnostics, etc.) or techniques with large amounts of data (e.g. NGS, microarrays) are easy to transfer into the LIS. It is clear, but in some laboratories not self-evident, that classical analytical machines can transfer their data to the LIS (e.g. if the LIS does not support this channel/connection of the analytical machines). Also clear but not self-evident for some LIS is the administration of laboratory routine operations like the monitoring of the analytical devices, the generation of work lists, the administration of measured values and controls, a control journal with automatic alarm by off-limit condition, a technical validation of the measured values, the lot documentation of the reagent, the administration of external quality control (round robin testing) and other items. If a LIS cannot fulfill these requirements, it is not future-oriented and improvements must be done by adding special program components.

In the section of the post-analytical phase, special programs are needed in data preparation and presentation, especially for test systems with large volumes of data (e.g. NGS, “omics” data). Because of the increasing number of long-term patients or patients with a preventive medical check-up, different forms of medical reporting (e.g. cumulative reporting, graphic course) are necessary. Furthermore, adapted medical reporting is needed, e.g. in pathology or transfusion medicine for patients with special antibodies as well as structured data transfer in the microbiology in special of results from resistance testing against certain antibiotics. In this context, it is necessary to link previous reports and measurement data. These reports and data must be provided by the LIS or specialized programs. When the medical report is finished, it must be transmitted to the entries/patient. New software for a secure electronic transfer is needed. Paper-based transfer should be restructured and easily readable. Finally, the medical reports/measured values must be digitally archived, independently from the data format of the LIS so that it can be reread if a change occurs in LIS. This is also necessary for legal reasons, because reports have different retention periods (amongst other things depending on the analyses performed) and must be stored 1:1 as delivered to the customer.

Moreover, special input screen for results from hand testing with manual data input should be programmed. These screens should ensure that the measured data is input twice to reduce transcription error.

Similar to current LIS, future LIS should not only have the possibility of manual validation and delta-checking but use AI for clinical autovalidation. Special software for archiving of samples is needed (this is also the last step of sample tracking).

At the stage of post-analytical management, web-based medical reporting and data retrieval are more commonly applied. This is only possible if the data security is supported by appropriate software in the background. From entries/clients, the transfer/print should be acknowledged and registered by software.

For data transfer, the communication between different data systems (e.g. different laboratories, hospital and doctor’s office) must be ensured. So far, many problems and client complaints have resulted from deficient transfer. As described in the section pre-analytical management, all communication between different data systems should be based upon master codes.

In laboratories, a control system of medical reporting should be installed. By using an enveloping machine, each printed medical report should be coded so that the machine can read it. While enveloping the machine should report each read ID to the LIS. Thus, the LIS would have a controlling mechanism to check if all printed documents are really enveloped.

Data/document storage has been mentioned at different stages. In general, an additional document
management system (DMS) should recover and store all ordering/reporting/measurement data and register data export. In addition, a DMS should not only save the unchanged/original measured values but also store background information such as reference values of the test that was used to analyze the sample as well as the method (this is of interest if the method has changed or another measuring device is used). Additionally, the software should tag the data for follow-up evaluation and for an easy change of the LIS provider. The actual standards for medical data telecommunications should be checked for improvements. For the sake of completeness, specialized software should be responsible for the different forms of billing and security backup.

Often it would be helpful or necessary to know about the medical history of a patient and be able to access this data for further medical evaluations [18]. The data of a patient that are electronically stored, called electronic health records (EHRs), were not designed as lifetime records and also not for exchange between different institutions [19]. So whenever patients change their doctor or consult a specialist they have different EHRs stored in multidata silos of different organizations [20]. They actually also lose the access to their patients’ history because the provider and not the patient gets the primary stewardship of the EHRs. In times of internet banking and social media, people are more and more used to internet interaction and trust in banking applications where they enter their pin code to make a bank transfer. People are also willing to provide their medical history to their actual doctor in charge to get best medical advices. An interesting approach here is to use the blockchain technology – best known in connection with bitcoin – to store medical data in the chain [21, 22]. The fundamental technology of blockchain is to store data decentralized on different computers (nodes) within a computer network [22]. Whenever new data shall be stored in the chain the nodes help to proof the correctness of the new data and help to make sure that all data that were ever stored in the blockchain are correct and manipulations are impossible. Such a system must clearly follow the rules of the GDPR especially when processing sensible data like personal or health data.

Azaria et al. suggested using blockchain to provide medical data access and implemented a prototype “Medrec” [23]. Some might feel anxious about personal healthcare data being stored in a computer network. Nobody wants his EHRs to be unencrypted and freely accessible. Medrec overcomes these worries by using a special kind of blockchain – ethereum. Within the Medrec implementation of ethereum, block content represents viewership and data ownership permissions that are shared by a private peer-to-peer network. This means Medrec is not directly accessible through, for example, the internet, it is only possible to access data stored in the blockchain for participants of this private network. All data are stored encrypted because on the ethereum blockchain a cryptographic hash algorithm is included to ensure health records from tampering and guaranteeing data integrity [24]. Ethereum uses programmable “smart contracts” to change viewership rights or to anonymize data for legal reasons. Smart contracts are small programs that can be added to data that are stored in the blockchain. For example, if anonymization of a blockchain dataset is necessary after a period of 10 years then there is the possibility to create a corresponding program, which takes over this task exactly. This program will be automatically executed by the Ethereum Virtual Machine (EVM). This is the “place” within ethereum where smart contracts run. Providers can add new medical data to the blockchain and patients can allow to exchange their personal medical data between doctors. Whether and who may access records is also governed by smart contracts. Here a smart contract could be implemented so that access to a record will only be withheld if both, the patient and the attending doctor, agree.

In the section of general requirements, universal tools and programs can complete the data management. Such tools, for example, can search the internet for the lowest price for controls (cost controlling). Merchandise planning and control tools can analyze the ordering of consumables. With the knowledge of the stock levels, it can inform the user to order new material in good time. Safety standards (e.g. biometric registration by the demand of measurement results, registration of the laboratory assistants by card or key) must be strictly adhered to by specialized software. Help tools should guide the user. Special software tools or the LIS itself must ensure that every change of the measured values, the annotations/interpretations (or other changes), even the opening of the data sheet is registered by a username. National and international standards must be controlled and documented with specialized software. In this context, it should be mentioned that the quality management (QM) programs must be restructured and extended. Document management and control systems for standard operating procedures (SOPs) should be improved or optimized. In this context, new (automatic) actualization system of the SOPs may be developed (e.g. automatic actualization of the reference range or the sample material) by changing the test. Customer relationship management (CRM) programs must be fitted for laboratories and their requirement analog to health promotion centers [25]. These programs are important tools.
for field workers and their communication to the laboratories and entries. Further documentation tools are very useful (with the possibility of documentation at every step of the complete analytical process), but every comment to a special case or a special issue must be in one data set and easily findable and retrievable (complete documentation of changes and retrievals by username). The results of methods of inspection must be integrated.

A further step is the implementation of training and qualification systems. This means the user will be trained and tested by software if he can master a certain process step (acquisition of a certificate of competence).

For a customer and quality oriented system, business intelligence (BI) and reporting is helpful. During this process, large amounts of data (e.g. from CRM, cost controlling, measurement data) will be collected, systematically analyzed and presented. BI is used in hospitals to optimize their processes and communication [26].

Especially for the section of general requirements, latest technology trends should be integrated. It should be possible to use highest security standards, e.g. by identification with a radio frequency identification (RFID) and monthly new activation of RFID by a user transmitted transacting number (TAN) [27]. The idea is that laboratory personnel would log on to thin clients (only operating system) by using RFID chips. To log on to LIS still a personal login and password should be necessary. To ensure that only authorized personnel can log on to the clients, regularly activation is necessary. The benefit here would be that several laboratory staff could use the same clients but each user would operate within his/her own profile and instantly switch between different user profiles.

Overview of the current programming of LIS as well as new developments and concepts for programming

Current LIS – monolithic systems

As already mentioned, most of the actual LIS have their origin in the years before 1980. In these times, software was mostly a monolithic application installed on a single server. Monolithic application means that all implemented features and programming code are packaged into one single application. A benefit for doing so is that the complete source code can be compiled and tested on one server and in most cases all features and functionality use the same database as well as common/global functions. Monolithic applications share the same memory and resources. User access to the monolithic application was mostly provided by clients. The backup concept often consisted of producing a copy of the main application server typically with daily data updates.

The original style of this programming architecture had many disadvantages, for example, compiling of the whole source code including automatic testing was very time-consuming. Often this process took several hours. When errors occurred during the compiling process or when automatic tests failed, the process had to be restarted all over again. That was one reason why the whole application was not compiled frequently and new versions were released once a year considering the need for a new feature. The software provider then, if this change required a new compilation of the source code, was only able to install it within the next release in a few months may be year(s). In most cases, the first chance to test the new-implemented feature(s) was when it was installed on the customer’s server. This means beta testing in a production environment and often features were not developed the way the purchaser wanted it to be. Customers need shorter development and release cycles and need to be involved more in the development process. There must be interaction during the development process between developers and users to prevent misunderstanding and developing features that the customer did not instruct or meant different. This last point is very important, because software development is never finished and new features have to be implemented to optimize the laboratory workflow or for legal reasons. So whenever this was necessary over the years, new features were only added by inserting further lines of code to the existing software, compiling and building a new, normally more complex, monolithic application. A possibility to overcome the problem of creating a big monolithic application was to write modular software. Modular software means that a monolith is divided into different components that work completely independent from each other and do not interact. The different tasks of a LIS are assigned to different modules and each module is developed independently from the other modules. This means in most cases the development of a new feature concerns only one module and all other modules are not affected. The single parts of a modular software are normally compiled fast and functional tests also only concern a single module. Often, the costs for a complete reengineering, for example, to establish a modular architecture were very high and hence not spent.
Service oriented architecture, particularly microservices

A relatively new option to face the problem of monolithic applications is to introduce a modern software architecture style known as “service oriented architecture”, for example microservices [28, 29]. Microservices are small and specific applications that work together to fulfill all tasks that monoliths would do as one big standalone software. Each microservice provides functionality (services) that other programs, for example, other microservices can use. The main difference to the modular software architecture is that microservices are able to interact with each other by exchanging messages.

A great benefit of dividing the whole functionality into small tasks (that are provided through microservices) is that whenever changes to software functionality have to be made, not the whole application has to be compiled, tested and newly installed but only those applications/services that provide this feature while all other microservices may remain as usual. This simplifies the process of software engineering because only a part of the software (only a microservice) has to be rebuilt. On the other hand, and to be honest, the complexity concerning the interoperability between the different services grows and shared memory access (that monoliths use) is faster than inter-process communication as used by microservices where each service is an extra process often on different servers. But the time difference for inter-process communication is with the help of modern computer systems and networks negligible. So the benefits of introducing microservices for systems in an ongoing development process overcome the usage of monoliths. How much communication between the different microservices is necessary depends on how a monolith is divided into different services. In order to be able to realize appropriate microservices, corresponding specialists are needed. Often LIS manufacturers do not have these specialists in their staff.

Agile software development: scrum as an example

After the discussion of the reorganization of a LIS from monolithic application to microservices, a new possibility to update a LIS should be presented. Agile methods such as scrum help to change the art of interaction between customers and software developers and accelerate software delivery [30]. Here, the most important aspects of scrum are explained to give a short overview. In scrum, software development is an iterative, agile process typically divided into sprints. A sprint is a time frame of given length where new features are developed. The typical length of a sprint is between 2 and 4 weeks. The aim of each sprint is to produce a software (increment) that is potentially releasable. Whenever a sprint ends, the development team presents a new version of the software but there is no requirement to release it. The great benefit is that customers have a chance very early to give feedback on the development result and if necessary can give advice to the developers and specify the requests more clearly. In scrum, there is no project manager but a development team (TEAM), a scrum master (SM) and a product owner (PO). The PO is responsible for the product. He or she has a vision of how the product should look like and what functionality it should provide. He describes his requirements in the form of user stories. It is very important to describe for whom (what kind of users/roles) the requirements shall be implemented and what benefit they will bring. This information is crucial for the developers because they get background information about the usage of their software and they are able to make better suggestions on how the software should be implemented. As a last step, the PO decides the prioritization of the story and the complexity in the form of story points. Story points are an indicator of the potential development time of a user story. To point out is that the PO only makes stories that can be completed within a sprint. This means they should not be too complex. If a story is too complex, it is better to split it into a few small stories.

The usual template for a user story looks like this:

As <role> I need <my wish> to <receive benefit>.

The whole TEAM is responsible for developing these requirements. The typical size for the TEAM is between three and nine people. The TEAM should have all skills needed to develop (and test) the requested software (increment).

The usage of the scrum methodology may help to convert current monolithic LIS software to a new kind of LIS that is built in a service oriented architecture. First steps should be to rebuild actual functionalities of a LIS by different microservices. This helps to reduce the complexity of the monolithic LIS step by step and transform the actual LIS into a central lean application with lower functionality. The central LIS is still the most important component in a distributed system of microservices and must provide application programming interfaces (APIs) that can be used by the newly introduced microservices.
Prospects

In all industries, expert systems are already established and will get more and more influence for decision makers. In 2017, Roche made a great step to better advise laboratory leaders by taking over the company Viewics Inc.

Viewics has a lot of expertise in BI and data-driven decision-making. Another sector that is going to become more important is telemedicine. In future, patients and doctors will communicate more and more through electronic devices and video chats. This will reduce time (for patients and doctors) because travel and waiting times will be reduced.

Based on the illustrated tasks of a future LIS, we think this is only solvable by a new concept, laboratory data management (LDM) as duties and responsibilities cannot be solved by one huge and closed-system LIS. A LDM should be a leading system, which coordinates specific programs for different specialized tasks (connection of useful tools and data transfer to and from them). It has to process and provide laboratory data for specialized programs; other programs have to process and provide data for the LDM. The LDM is a modular system with an individual adaptation to the laboratory (e.g. small, highly specialized laboratory or laboratory with a high throughput). In addition, it is advisable for high flexibility and change of programs that the data (measurement data, medical reporting and master data from the LIS) are saved in a form, which can easily be read by another software.

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: None declared.

Employment or leadership: None declared.

Honorarium: None declared.

Competing interests: The funding organization(s) played no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the report; or in the decision to submit the report for publication.

References