

Agile Software Development for Vaccine Delivery in The Context of the Coronavirus Pandemic

Alexandru Ionuț NIȚU
Academy of Economic Studies, Bucharest, Romania
alexandru.nitu@csie.ase.ro

Agile software development is a group of project management methods that involve identifying customer requirements and developing solutions collaboratively. It is based on iterations of progress that consist of functional intermediate variants with few software bugs. Emphasis is placed on the frequent dialogue between team members through daily meetings where the progress made the previous day and the problems encountered are discussed. A customer representative frequently participates in these discussions. The Coronavirus pandemic has changed everyone's lifestyle. The only possibility of returning to a normal life seems to be the appearance of a weapon against the invisible enemy. Time plays the most important role, and errors must not exist. In just two weeks, a logistics software system was implemented within the National Institute for Medical-Military Research-Development "Cantacuzino" in Bucharest, to support the effort to combat the disease.

Keywords: Agile Software Development, Covid-19, Scrum, Inventory Management, Traceability

DOI: 10.24818/issn14531305/25.3.2021.05

1 Introduction

A mysterious disease afflicts several patients in Wuhan, China. Although doctor Li Wenliang, recognizing the symptoms and how the virus is transmitted, warns that we are facing a pandemic, it is ignored, and the effects are catastrophic. The pathogen spreads rapidly and makes more than 100 million people on the planet sick, with more than 2.5 million losing the battle with the disease. Quarantine is established in turn in most countries for different periods of time, and after it can no longer be maintained, it is replaced by restrictions and social distancing. The situation is getting worse, and it is now becoming a health emergency. Prevention limits the spread. Global travel restrictions are imposed that are shaking financial markets. In hospitals, doctors and nurses are assailed by the wave of patients with symptoms of viral pneumonia who do not respond to regular treatment. One by one, the hospitals are filled

to the brim with patients. In Europe, the first and, by the way, the most affected country is Italy. From there, the infection spreads and takes over all states.

The situation creates perhaps the biggest challenge so far. Despite these restrictions, the expansion continues, and it seems that the only weapon in the fight against the invisible enemy can only be the vaccine. The first vaccine, produced by Pfizer Inc. in collaboration with BioNTech receives the approval of the Food and Drug Administration of the United States of America on December 11, 2020¹. Then followed shortly (December 18, 2020) the solution offered by Moderna TX Inc², respectively AstraZeneca (January 29, 2021)³. The logistical challenge of transporting hundreds of millions of doses of vaccine to all corners of the globe is already anticipated.

In Romania, the first doses on December 26, 2020, at the National Institute for Medical-

¹ Pfizer-BioNTech COVID-19 Vaccine, <https://www.fda.gov/emergency-preparedness-and-response/coronavirus-disease-2019-covid-19/pfizer-biontech-covid-19-vaccine>

² Moderna COVID-19 Vaccine, <https://www.fda.gov/emergency-preparedness-and-response/coronavirus-disease-2019-covid-19/moderna-covid-19-vaccine>

³ COVID-19 Vaccine AstraZeneca, <https://www.ema.europa.eu/en/medicines/human/EPAR/vaxzevria-previously-covid-19-vaccine-astrazeneca>

Military Research-Development "Cantacuzino" in Bucharest, Figure 1. It is expected by a team of doctors, nurses, and IT specialists. The rules of handling and storage are among the strictest and most difficult to comply with. To cope with the rigors of the Army and to accommodate all the particularities of the different types of vaccines, a logistics software system was developed. It was used from the first reception and in all deliveries that followed it.



Fig. 1. The first doses of vaccine provided by Pfizer

2 Agile principles

In a world of software projects developed using the *Waterfall* methodology, in which a product is developed in different stages, with distinct stages of analysis, development and implementation, a new work process is required. It must improve the speed of development, provide predictability, and provide users with a product that they want and pay for [1].

The set of *Agile* principles provides directions in identifying the most appropriate methods and procedures for the development team to be as agile as possible. It values the importance of people above processes, the development of functional products to the detriment of documentation that shows how

the software should work, collaboration with customers to the detriment of negotiations and the response to change to the detriment of a plan.

Agility consists in continuous adaptation and the ability to permanently improve, evolve, and adapt the way you work [2]. This played a very important role in the whole implementation process if we consider the very short time in which the project was implemented (two weeks for Pfizer and two weeks for Moderna). Scrum is a working methodology, a framework that provides development teams with ways to organize and improve both work speed and product quality. They are based on identifying the reasons why each work task lasts too long. After an initial discussion with the decision makers, the future functionalities of the software system were fragmented into small and individual tasks. Each of these was estimated as time and value brought to the project. The estimates made by the team offer a much higher accuracy than the individual ones. These are not done through discussion, as the opinion expressed by the first person will influence the others as well. While one person is listening to another, he, or she may think that certain things that he or she does not consider to be right are in fact correct. Thus, all people come to agree with an opinion, although each of them has a reservation. In literature, this effect has been explained according to the information cascade.

The whole process was fragmented into two sprints for each supplier. The development team selected the most important tasks for the first period and, every day, meetings were set up to discuss the progress made the day before. Sushil Bickchandani, David Hirshleifer, and Ivo Welch are the authors of *A theory of Fads, Fashion Custom, and Cultural Changes as Informational Cascades* in which they describe human behaviour as follows: *An information cascade occurs when it is optimal for an individual, after observed the actions of those in front of him, to follow the behaviour of previous individuals, without taking into account his own information.* [3]

A dashboard that contains the *backlog* and the

progress of the project was built. It contains five panels. The first one contains the *backlog* and represents the list of tasks we must complete. Three other panels display the list of tasks the team must complete in the current week, the list of tasks allocated to users and the list of tasks that are completed. The last

one shows the number of accumulated points at the current time. Any employee of the company can read the dashboard and know the progress. The Figure 2 contains a sample of the dashboard. It is built in *ASP.NET* using *DevExpress* components.



Fig. 2. Scrum dashboard

The most important aspect of this methodology is the fluency that the development process must have. The professionalism of the development team is based on more than fifteen years in consulting and development of software applications for logistics. During this period, the impediments in development were observed and, one by one, eliminated, these tasks falling to the management of the company. Thus, the process of configuring applications and developing new features went smoothly. The regular meetings with representatives of the Cantacuzino Institute, the Ministry of National Défense and the Ministry of Health aimed to present the real progress. During them, the functionalities developed along the way and the necessary changes to be made in the next period were discussed.

3 The software system

To cover the entire operational flow, the proposed solution consists of three applications: the *front-office* application, the *back-office* application, and the reporting

application (Fig. 3. Software system architecture). The chosen architecture is modern, on three levels: client, server, and database. The client application can be any device on which the user is running applications. In this case, the client applications are *front-office*, *back-office* and reporting and execute various requests to the server. At the same time, it receives, interprets, and displays the answers received from the server. The cloud application receives all requests from client applications, interprets them [4], executes them over the database and responds to requests. It is hosted on an *IIS* server and is developed in the *C#*, *AspNet* programming languages. At this level, the code instructions assemble the response for the client by interacting with the database. Such requests are sent when the user, for example, wants to consult the list of commands: the Show commands button is pressed, and the cloud server turns the request into an SQL request that it sends to the database server; the latter extracts the data from the table and transmits it to the server

application from where it is propagated in response to the requester.

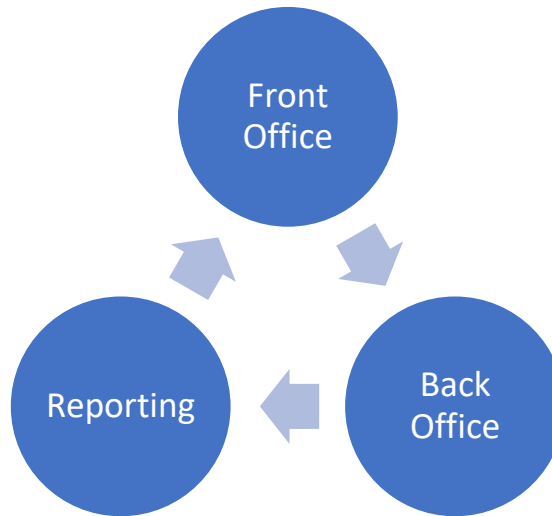


Fig. 3. Software system architecture

Information storage corresponds to the third level of architecture. The layer is represented by the database hosted by the Microsoft SQL server. The database is queried using the *Transact-SQL* programming language based on the SQL standard. There are three types of requests: for data manipulation, for manipulating the database structure, and for access control. The first category consists of the instructions *SELECT*, *INSERT*, *UPDATE*, *DELETE*. Modifying the database structure is done using the *CREATE* (for creating new objects), *ALTER* (for modifying existing objects), *DROP* (for deleting existing objects) instructions. For the allocation or revocation of access rights to database users, the instructions *GRANT* (for allocation), *REVOKE* (for dislocation), and *DENY* (for prohibition) can be used. [5]

The script below, extracts the following task the current user must execute in the current warehouse. It queries three tables: the tasks table, activities associated with persons and activity details. The parameter *@PersonId* specifies the unique identifier of the person for which the database is queried, while *@WarehouseId* specifies the unique identifier for the warehouse in which the person currently works. First, the script checks all the tasks available in the same warehouse (*idAdmWarehouse = @WarehouseId*), if there is already a task assigned to the person

(*@idAdmPersonExecute = @PersonId*) or is not assigned at all, the task is not yet completed nor cancelled (*idTaskState < 5*) and if the person has assigned the necessary activity for the task to be executed (*tblAdmPersons_AdmActivity = @PersonId*). After the *WHERE* is executed, the list of tasks is available to be sorted and to extract the most important task. A few sorting principles are specified in the *ORDER BY* clause. First criteria is the priority of the order. The lower the priority, the more important it is. The priority can be set manually by a user or by the system, respecting different criteria. A priority less than twenty makes the order to be prepared as soon as possible. The second item is the priority associated with the activity. One person can, for example, work both in reception and at delivery. To make a difference between them, a priority at activity level is used. If the coordinator wants the person to work in reception but also help in delivery if the person doesn't have work to do, the coordinator can assign the delivery activity with a bigger priority. Next it is the task's priority, then the first in-first out rule for the orders and for the tasks. The values returned are used by the *front-office* application to display to the user the necessary information for reception, delivery, or other warehouse process.

```

SELECT
    tblTasks.id,
    TaskGroup,
    idAdmPersonExecute
    PersonExecuteName,
    idTaskState,
    idTaskType,
    TaskTypeName,
    tblTasks.InsertDate,
    tblTasks.OrderNumber,
    IIF(tblTasks.Priority <= 20, 0, 1) AS [Prioritate comenzi urgente],
    tblAdmPersons_AdmActivity.Priority AS [Prioritate activitate
persoana],
    tblTasks.UDF_FLOAT_1 AS [Prioritate in functie de ora de livrare],
    tblTasks.Priority AS [Prioritate task/comanda],
    tblTasks.idOrderOut AS [ID comanda iesire],
    tblTasks.InternalPriority AS [Prioritate interna]
FROM tblTasks INNER JOIN
    tblAdmPersons_AdmActivity ON tblTasks.idAdmActivity =
tblAdmPersons_AdmActivity.idAdmActivity INNER JOIN
    tblAdmActivities ON tblAdmPersons_AdmActivity.idAdmActivity =
tblAdmActivities.Id LEFT JOIN
    tblTaskOrders ON tblTasks.Id = tblTaskOrders.Id
WHERE tblTasks.idAdmWarehouse = @WarehouseID AND
    (idAdmPersonExecute IS NULL OR idAdmPersonExecute = @PersonID OR
@PersonID IS NULL) AND
    idTaskState < 5 AND
    (idMasterTask IS NULL
    OR
    EXISTS(SELECT 1 FROM tblTasks AS masterTask WHERE masterTask.Id
= tblTasks.idMasterTask AND masterTask.idTaskState >= 5)
    OR
    NOT EXISTS(SELECT 1 FROM tblTasks AS masterTask WHERE
masterTask.Id = tblTasks.idMasterTask)) AND
    (tblAdmPersons_AdmActivity.idAdmPerson = @PersonID OR
tblAdmPersons_AdmActivity.idAdmPerson =
tblAdmPersons_AdmActivity.idAdmPerson)
ORDER BY
    IIF(tblTasks.Priority <= 20, 0, 1),
    tblAdmPersons_AdmActivity.Priority,
    tblTasks.Priority,
    tblTasks.idOrderOut,
    tblTasks.InternalPriority,
    tblTasks.Id

```

In the reception process, barcode scanners are used, facilitating the entry of data into the system. This mechanism is used by the staff of the Institute in the reception and delivery activities. Mobile terminals, Figure 4, are dedicated to logistics operations, being built to withstand the harshest conditions. The application for this process is developed using *Xamarin* technology that allows writing in the *C#* language and compiling for Android devices. It is permanently connected to the server application to exchange data.

The addition of incoming orders sent by suppliers, deliveries to vaccination centres across the country, batch management and traceability are managed by the back-office application.

The software is designed using *Windows Forms* technology in the *C#* language. To be as easy to use as possible and to have a visual appearance as pleasant as possible for the users, *Dev Express* controls were used. They make up the entire graphical interface of the back-office application. The reporting module

confirms compliance with all rules for handling and storing vaccine doses, regardless of the supplier, and provides accurate information on their full traceability, from receipt to delivery. Each of these three modules connects to the centralized server application. It is based on web services.



Fig. 4. Mobile device used in logistics processes

Users are logged in to any of these applications using personal username and password accounts. Each action registered in the applications is accompanied by the name of the user who performed it. Thus, the activity of all users is known.

4 Reception

The receive process involves the existence of an input command added to the system. It is created by the users of the *back-office* application in the corresponding windows according to the information received from the provider. The application also allows semi-automatic import (from text files) or fully automatic (via *Electronic Data Interchange, EDI*, such as *Application Programming Interface, API*). In this case, the provider's software system and the Institute's software system must be connected to exchange information. The representatives of the two software systems must establish the way of communication between the two systems, to develop the interfaces, finalizing with the effective development. This

automation reduces the operating times in the application, which also leads to a decrease in human errors.

For complete traceability, an input command goes through several states. When it is created, the status assigned by the system is *Added*. At this stage, changes can be made to it, such as changing the quantities, adding additional products, etc. The order remains in this state until its details are confirmed, at which point it enters the *Confirmed* state. An order can only be confirmed if it is in the *Added* state. To start its actual processing, the command must be *Validated*. An order can only be activated if it has been previously confirmed. This action is taken by the users, manually, when the truck arrives for reception. From now on, the order is available on mobile terminals for processing. Users can start the actual receipt of products from the supplier. Following completion of receipt of vaccine doses, the order must be physically and completely marked as fully received, an action manually triggered by users upon completion. This action moves the command to the *Completed* state. In exceptional cases, orders may also be cancelled. This event is marked by changing the status of the order to *Cancelled*.

The states in which a command is located are stored in the database in a table that has a *Many to Many* relationships with the command table. Thus, for each status change, a record is added to the table. It contains two given columns that make up an interval. The interval denotes the period during which the command was in that state. When switching from one state to another, the time interval of the current state closes and another record is added, but which has completed only the beginning of the interval. The current state differs from the others by the blank value near the end of the interval. We know in this way when an order was created, how long it took to enter, when it was put into operation, how long it took to process and when it was completed. This information can be viewed at any time in the reporting application via existing views.

States are stored in database in a table called

tblOrdOrdersOutStates. It contains all the states the system can handle: *Added*, *Confirmed*, *Available*, *Closed* and *Deleted*. Each order can be at a moment in only one state, but during its lifetime, can be in many more. The *many-to-many* relationship its

stored in *tblOrdOrdersOut_OrdOrdersOutStates* table. The representation of this tables is listed in Figure 5 (tables are simplified for easier understanding).



Fig. 5. Orders States Tables

Adding, confirming, and activating incoming commands is done through the desktop application by a *back-office* operator. A stored procedure is used to make the changes in the database. First, it checks a few details about the order, like it cannot be closed only if was

picked from shelves and loaded in the truck. This assures that the order is not marked as finalized in the database before it was actually finalized. In the following instructions, the current state is *closed* using the *UPDATE* listed below.

```
UPDATE dbo.tblOrdOrdersOut_OrdOrdersOutStates
SET [DateStop] = GETDATE()
WHERE idOrdOrderOut = @OrderOutID AND
DateStop IS NULL
```

The change of state is done in table *tblOrdOrdersOut_OrdOrdersOutStates* by update the *DateStop* column with the current date. The *GETDATE* function returns the current date of the server in *DATETIME* format. The *WHERE* clause from this statement is perhaps the most important, because it *finds* the only open state of the order, the current state, the one that does not have and ending at the moment (*DateStop IS*

NULL). *@OrderOutID* parameter contains the unique identifier of the order. An *INSERT* statement adds a different record of the newly state in the same table. It fills the current state in column *idOrderOutState*, the current date and time in column *DateStart*, the unique identifier of the person who makes the change in column *idAdmPerson*, leaving *DateStop* column blank.

Next, the activity moves out of the offices,

into the reception area. The staff of the Institute, using the *front-office* application, receives the products. With the physical completion of the takeover of the goods, it confirms in the application, an event that causes the change of the order status in *Completed*. The physical reception consists in the handling of the Soft Box, Figure 6, from the supplier (Pfizer, Moderna, AstraZeneca), their sale and the effective handling of the vaccine doses to the storage spaces. In this process, special scanning head devices are used to read the barcodes written on the transport boxes. These automatic readings reduce the time when vaccine doses are not in the optimal storage range (for Pfizer, between -80°C and -60°C , for Moderna at -20°C , and for AstraZeneca between 2°C and 8°C).



Fig. 6. Soft Box transport

The traceability of goods in a storage space always requires the knowledge with maximum accuracy of the place in the warehouse where each unit of goods is located. Thus, any move of any kind must be performed through the software application. The traceability of a product arises in the process of receipt. On each box received from the supplier, a small label is affixed containing an integer, positive and unique number

represented by an appropriate barcode. Receipt consists of scanning this code and entering the number of doses in the container. From this moment, the stock is loaded with the respective items. Moving this box to one of the refrigerators means moving from a receiving location to an Institute storage location. Thus, the reception management is relieved of those quantities, while the warehouse management is loaded by them. The label applied to the box is to be scanned in other processes performed in the warehouse, such as delivery. In this case, a step in this action is to scan the box from which doses are taken and enter the amount.

To shorten the time of adaptation to the new software application and to facilitate the actions of users, the front-office screens have been built respecting the principles of ergonomics and simplicity. At its core is a system of work tasks that are generated by the brain of the whole. They are transmitted one by one on mobile devices according to the last action taken by the user. Confirmation of the execution of the current task is done by filling in the various fields in the application. The next task displayed by the device is in accordance with the algorithms introduced in the application based on the logical schemes designed during the analysis period.

The display of work tasks on mobile devices is done on lines, each containing some useful information for the user. In Figure 7, the last step in a vaccine reception is displayed. On the first line displayed, there is the task number, 667 generated so far by the system, and the title, Rec - 4/4. To assist the user in completing the operation, information about the current order is included. Specifically, the order number is displayed, the eleventh order added so far in the system, the supplier, Pfizer, the quantities to be received and the current box. Each of these fields is marked in blue and is called Info.

The actions taken by the user to successfully confirm the activity are to scan the refrigerator in which to place the cans, the box from which they are taken, the box contained in the box and the item. To mark distinctly the fields for which the user must take an action are

underlined in yellow and are named Confirm. After pressing the Ok button, the task is performed, and the items are scripted from the box to the scanned refrigerator. Using this mechanism, the time required to move Pfizer doses from the transport boxes (where the temperature is -65°C) to the refrigerators is about 60 seconds, which is one third of the maximum time allowed by the supplier.



Fig. 7. Task in the reception process

5 Delivery

The Cantacuzino Institute is the National Center for the Collection and Distribution of Vaccine Doses, being the only institution that has equipment capable of reaching a storage temperature of -80°C . They are kept in the institution's warehouses for as short a time as possible distributed to all vaccination centres in the country. The exit from the management of the goods is done through the delivery process. Like the reception process, the delivery consists in the creation in the software system of exit orders by the back-office operators. They centralize requests for vaccine dose quantities and, depending on stocks and delivery availability, organize the

distribution of the next period. Activating exit commands allows users to collect vaccine doses from freezers. And the actions of this process are performed through the software system, being displayed on mobile devices through successively displayed screens. In this activity, time plays a very important role. Every moment is monitored and recorded in the application. A box in which vaccine doses are packaged may be handled only once within two hours. For Pfizer, the maximum time allowed to work with a box is five minutes and three minutes if the doses are moved out of the boxes. From the time of removal from the freezer for commissioning, there is a different shelf life than the shelf life. This shelf life calculated from the time of preparation for delivery is printed on the collective packaging label. Labels are automatically printed from the application as a step in the delivery process and is useful for doctors.

Figure 8 shows an example of scanning a dose of vaccine to be delivered. The first two lines of the screen are like the reception process. In the Info area, there are two other fields, the location from which the products must be taken (the top of the refrigerator number 2), respectively the name of the item to be delivered. The Q Com field displays the progress made so far in preparing the order in number of boxes (current versus total) and pieces (bottles). The red colour of the writing informs the manager that the delivery does not require a full box of 195 doses and that the number of doses indicated in the activity must be opened and extracted. The handling time indicated by the manufacturer for such cases is only three minutes, compared to 5 for whole boxes. This colour change is a small detail, but a very important one for the daily activities of the staff.



Fig. 8. Task in the delivery process

The Box Code field is the first that the user must scan, being represented by the label applied on the box during the reception process. If the quantity is smaller than the one in the box, it is out of stock. If it is equal to the available, the box ceases to exist (it is deleted from stock) but continues to exist in the warehouse. An error is displayed if the user enters a larger quantity than the one in stock. The Item field is filled in by scanning the barcode received from the manufacturer. The quantity is entered manually by the operators by pressing the corresponding numeric buttons on the keyboard of the mobile device. If a box containing fewer units is scanned than is required to be delivered to order, a new load is automatically generated for the remaining unprocessed quantity. The user continues by scanning another box and entering the amount he takes from it. The Destination Box field is the last scan in this delivery step. Like the reception process, labels containing integer, consecutive and positive numbers and their barcode are applied. The software system

manages the generation and use of these tags and ensures that they are used uniquely. By scanning the label in this step, a new, mobile storage space is created in the delivery management. The delivery process consists of moving the product units from a storage location to a delivery location.

All work tasks required to process a delivery order are generated automatically. If the finality of the process is not known a priori during the reception (the theoretical, written quantity of products to be received is known, but the actual quantity received from the supplier is not known), the final delivery of an order consists in the complete preparation of quantities. Thus, the output commands are closed automatically by the software system when the quantity prepared by the managers is equal to the written quantity on the order. There is also the possibility of forcibly closing an exit order even if the quantities have not been prepared in full. There is a possibility that an order may not be fully honoured due to, for example, lack of stock. In such situations, a back-office operator triggers the forced, early termination action.

6 Reports

In a context where vaccine dose storage temperatures are extremely important, assuring medical staff that storage times at temperatures other than mandatory is extremely important. The value of information has become directly proportional to the speed at which it can be accessed in the current economic context. Populating reports with appropriate data involve identifying metrics and indicators relevant to this activity. The front-office application records all movements of goods from transport boxes to freezers, from freezers to refrigerators and from refrigerators back to transport boxes. It is associated with a time in which each of these actions was performed. Mathematical calculations demonstrate compliance with all vials in optimal conditions. These results are displayed via the reports defined in the application Figure 9 shows a few records for moving doses into or outside a freezer.

Nr. comanda	Persoana	Durata	Timp	Activitate	Tip de task	Ora aduagre	Ora inceput	Ora finalizare	Locu
OUT 0040	Andrei Petrescu	130	130	Receptie	Picking	13:39:40.3600000	13:39:40.3600000	13:41:50.7800000	10091
0044	Andrei Petrescu	65	65	Receptie	Picking	13:46:46.3600000	13:46:46.3600000	13:47:51.0033333	10091
OUT 0021	Anca A	93	93	Receptie	Picking	17:08:24.2300000	17:08:24.2300000	17:09:57.8233333	10130
OUT 0021	Anca A	12	12	Receptie	Picking	17:09:57.8300000	17:09:57.8300000	17:10:09.7766667	10130
OUT 0021	Anca A	26	26	Receptie	Picking	17:10:45.8400000	17:10:45.8400000	17:11:11.8866667	10130
OUT 0021	Anca A	36	36	Receptie	Picking	17:10:09.7833333	17:10:09.7833333	17:10:45.8300000	10130
OUT 0021	Anca A	32	32	Receptie	Picking	17:11:26.4800000	17:11:26.4800000	17:11:58.7366667	10130
OUT 0021	Anca A	31	31	Receptie	Picking	17:12:06.9766667	17:12:06.9766667	17:12:37.8166667	10130
OUT 0021	Anca A	8	8	Receptie	Picking	17:11:58.7433333	17:11:58.7433333	17:12:06.9700000	10130
0046	Andrei Petrescu	32	32	Receptie	Picking	14:25:08.4800000	14:25:08.4800000	14:25:40.7866667	10148

Fig. 9. Storage report

The delivery report contains information on the number of the consignment note, the boxes from which the doses were taken, the freezer (temperature between -80°C and -60°C) in which they were stored during the stay in storage, the time when it was opened (consisting of date and time in milliseconds), the time it arrived in the refrigerator

(temperature 2°C - 8°C), the time it was loaded into the transport machine (located in transport boxes with a storage temperature between 2°C - 8°C), the name of the destination unit and its address. The report accompanies each delivered order. In Figure 10 are listed a few deliveries to hospitals from Bucharest.

Task	Comanda	Boxa	Locu	Catig	Ingrader	Ingrader deactiva	CTM	Mutaria 2 #/s	Inventarie	Destin
9606	149E AV132E	11812	17740		09-03-2021 12:11:30	W87224002	09-03-2021 12:13:13	09-03-2021 13:10:10	COMANDAMENTUL LOGISTIC INTRUNJINT	
9604	1494 AV1327	11812	17739		09-03-2021 12:11:30	W87224002	09-03-2021 12:13:13	09-03-2021 13:09:14	REG. 306 MIHA VITEAZUL	
9601	1488 AV1326	11814	17738		09-03-2021 12:11:30	W87224003	09-03-2021 12:13:14	09-03-2021 13:07:10	CENTRUL 13 COMUNICATI SI INFORMATICA	
9600	1488 AV1326	11812	17738		09-03-2021 12:11:30	W87224002	09-03-2021 12:13:13	09-03-2021 13:07:17	CENTRUL 13 COMUNICATI SI INFORMATICA	
9598	1483 AV1325	11814	17737		09-03-2021 12:11:30	W87224003	09-03-2021 12:13:14	09-03-2021 13:04:49	Spitalul De Urgenta Prof Dr Dumitru G	
9595	1482 AV1324	11814	17735		09-03-2021 12:11:30	W87224002	09-03-2021 12:13:14	09-03-2021 13:03:54	Institutul National de Boli Infectioase	
9594	1482 AV1324	11814	17735		09-03-2021 12:11:30	W87224002	09-03-2021 12:13:14	09-03-2021 13:04:05	Institutul National de Boli Infectioase	

Fig. 10. Delivery report

A second very important report generated by the application provides information on the daily stock of vaccine doses. It displays for each type of the unique identification code of the box from the supplier, the number of doses received or delivered that day, the current stock and the necessary for the next day. The latter is calculated as the sum of the quantities of products on the exit orders existing in the system. The report is automatically generated by the system and sent every day via email to the decision-making staff of the Institute, the Ministry of Health and the Ministry of National Defense. To this end, an email box has been created so that it can be used by the software system.

These reports provide information essential to keep the business running on the desired trajectory and in optimal conditions. For each of them, the team of programmers defined views in the database. Views are objects that contain SQL queries. Reading data from a view means executing the underlying request [6]. They optimize the programming process, as they save development time by keeping the

most used applications in the application. From another point of view, they represent an added level of security to the software system. Using views instead of requests directly from the table offers the possibility to filter data both horizontally (excluding columns with sensitive information) and vertically (filtering records according to different criteria) [7]. Thus, the reports contain only information intended for the users who access them.

7 Hosting

Hosting the applications is another big concern. Each project has its own IT needs. Cloud computing has been chosen because of its scalability, simplicity, and security. Scalability is the process by which multiple computing resources can be allocated to an application. If the demands grow, additional resources are needed. A more appropriate choice than purchasing CPU, RAM or storage and installing in current systems (if they even support the growth) is the purchase from a third party. Since costs in the cloud are based on consumption, they will be lower than

hardware costs. The company which owns the servers also bear the costs of them, manage software updates and all hardware. [8] Security is probably the most important factor in choosing cloud infrastructure. In the age of ransomware attacks, an on-premises server is increasingly vulnerable, and its security costs are increasing. Because such a virus encrypts all data on a machine and is impossible to decrypt, access security solutions, such as firewalls and antivirus applications, are needed. However, new versions of such viruses are becoming more capable and more difficult to combat. Locky, WannaCry, Cryptolocker and Petya are just a few viruses that have spread to over 150 countries and caused huge damage. WannaCry alone has lost more than \$ 4 billion. [9]

Under these conditions, the restoration of infrastructure plays an extremely important role. First, equipment is needed to back up both data and configurations. Such equipment comes at a high cost and must be secured as well. In a case of force majeure, it is necessary to restore the infrastructure as quickly as possible, so that the business can continue its activity. However, restoration times can be hours or even days. In a cloud infrastructure, all these functionalities are normal and are offered as elements of the core packages. Microsoft Azure Figure 9 provides the ability to back up to the same data centre or to others owned by your company. They can be done as often as needed and even contain the status of the applications.

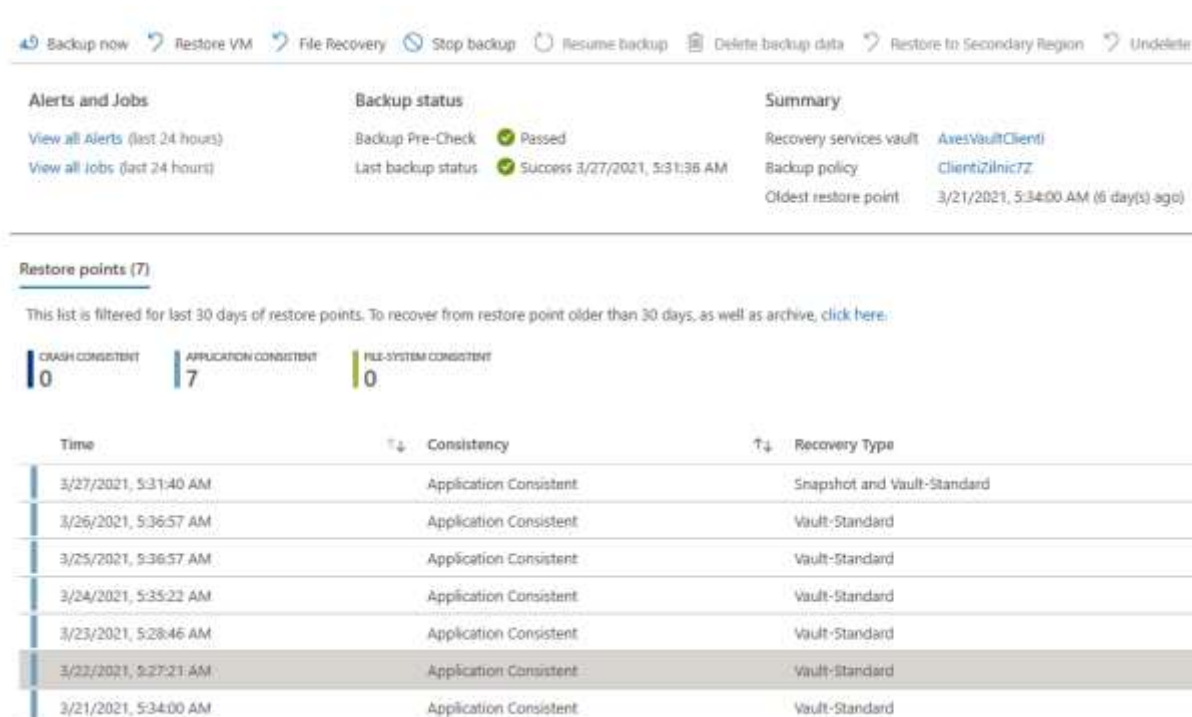


Fig. 11. Backups in Microsoft Azure

Thus, the restoration means not only the restoration of the data, but also of the applications with the necessary settings and exactly the state in which they were at that time. Moreover, individual files can be recovered as they existed at a certain point in time. [10] The chosen solution respects the rigors of the army.

8 Conclusions

The system is built to help the personnel so they can deliver fast and with maximum accuracy the so needed doses of COVID-19 vaccine. The system reduced errors in reception, picking, inventory and other processes. It contains tools for troubleshooting and identification so that no dose expires in the freezers. Real time reports

allow management to stay connected with stocks and delivery progress. The whole system is designed to be easy to use and gives to the Government the tools to coordinate the distribution. It helped delivering millions of dozes across the country, helping the personnel with documents, and optimizing their workflows.

References

- [1] J. Sutherland, *Scrum, the Art of Doing Twice the Work in Half the Time*, 2014 ed., New York: Crown Business, 2014.
- [2] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1st ed., Upper Saddle River, NJ: Prentice-Hall, Inc, 2002.
- [3] S. Bikhchandani, D. Hirshleifer and I. Welch, "A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades," *Journal of Political Economy*, vol. 100, no. 5, pp. 992-1026, 1992.
- [4] J. Rittinghouse and J. Ransome, *Cloud Computing - Implementation, Management, and Security*, 1st ed., Boca Raton, FL: CRC Press, 2010.
- [5] Microsoft Corporation, *20761C - Querying Data with Transacti-SQL*, Microsoft Corporation, 2017.
- [6] I. Ben-Gan, *Quering Data with Transact-SQL*, Pearson Education Inc, 2017.
- [7] P. A. Carter, *Securing SQL Server*, Botley, United Kingdom: 2016, 2016.
- [8] A. Velte, T. Velte and R. Elsenpeter, *Cloud Computing. A Practical Approach*, New York: McGraw-Hill Companies, 2010.
- [9] Kaspersky Lab, "Ransomware Attacks and Types – How Encryption Trojans Differ," 2020. [Online]. Available: <https://www.kaspersky.com/resource-center/threats/ransomware-attacks-and-types>.
- [10] Microsoft, "What is the Azure Backup service?," 24 04 2019. [Online]. Available: <https://docs.microsoft.com/en-us/azure/backup/backup-overview>.



Alexandru Ionuț NIȚU, with over 10 years of experience in software development in the field of logistics, graduate of the Academy of Economic Studies in Bucharest, master in Information Security, SQL Server trainer. He believes that education is the ingredient that can change the world and that is why he is dedicated to the permanent improvement of this field.