

Lappeenranta University of Technology
School of Engineering Science
Software Engineering
Master's Programme in Software Engineering and Digital Transformation

Mohammad Fakhrul Islam Farhad

**TOWARDS AN AUTOMATED DELIVERY PIPELINE FOR A
TYPICAL SMALL SOFTWARE COMPANY**

Examiners : Professor, Ph.D. Kari Smolander

Supervisors: Professor, Ph.D. Kari Smolander

ABSTRACT

Lappeenranta University of Technology

School of Engineering Science

Software Engineering

Master's Programme in Software Engineering and Digital Transformation

Mohammad Fakhrul Islam Farhad

Towards an automated delivery pipeline for a typical software business

Master's Thesis

53 pages, 5 figures, 2 tables

Examiners: Professor, Ph.D. Kari Smolander

Keywords: delivery pipeline, continuous integration, continuous delivery, DevOps, agile project management

Due to the advancement of IT infrastructure and easy availability of different IT services, a typical small organization, which provides some form of digital services to customers tend to rely heavily on its suppliers. Achievement of organizational goals and managing the key challenges are directly involved with the entire delivery pipeline starting from the requirement to delivery of the product or service. Hence a defined delivery pipeline is required. One of the objectives to have a defined delivery pipeline is to have enough control on the entire process, so that the process can be analyzed for further optimization such as introducing automated testing, continuous integration, and eventually reduce the

timeline from requirement to delivery. In this research I shall attempt to identify and capture the current delivery pipeline for PharmaServe, identify the enhancement opportunities according to the business need, and challenges it should manage for such implementation.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Dr. Kari Smolander who did not leave while I was continuously breaking the promises and taking a longer time than expected. He has guided me all the way to shape the study.

I am thankful to the organization, which has given me the opportunity to work on their systems and be acquainted with different organizational components.

Last but not least I would like to thank my lovely wife and cute daughter who supported me by creating an environment at home so that I can concentrate on the work, finish the things before summer and as a consequence we enjoy Finnish summer altogether.

TABLE OF CONTENTS

INTRODUCTION	7
BACKGROUND	7
JUSTIFICATION OF THIS STUDY	8
GOALS AND DELIMITATIONS	9
STRUCTURE OF THE THESIS	10
LITERATURE ON DELIVERY PIPELINES	11
RELEVANT LITERATURE	11
RESEARCH ENVIRONMENT AND METHODOLOGY	16
THE ORGANIZATION AND THE PRODUCT	16
THE DEVELOPMENT PROCESS	20
CURRENT DELIVERY PIPELINE	22
OBJECTIVES OF THE SOLUTION	27
RECOMMENDED DELIVERY PIPELINE	29
ENHANCE THE SOFTWARE CONFIGURATION MANAGEMENT PROCESSES	31
ENHANCE TESTING AND QUALITY ASSURANCE PROCESSES	35
TO-BE DELIVERY PIPELINE	40
DISCUSSION	44
CHALLENGES DURING THE CHANGE	45
LIMITATION OF THIS STUDY	46
CONCLUSIONS	47
REFERENCES	48

LIST OF SYMBOLS AND ABBREVIATIONS

ASR	Architecturally Significant Requirements
BPM	Business Process Management
CD	Continuous Deployment
CDE	Continuous Delivery
CI	Continuous Integration
SCM	Software Configuration Management
SDK	Software Development Kit
SDO	Software Development Outsourcing

1 INTRODUCTION

1.1 Background

Due to the advancement of different cloud based IT services, the initial burden of setting up a new business is waived out from the entrepreneurs. Recently a significant rise in software product centric organizations has been observed [1]. Easy availability of online service providers, flexible pricing policies of such services, competitive force from the peer business groups tend to impact the organizational value delivery chain. Mass participation in business resulted from easier access to the online services make the market more competitive, and an indirect pressure to reduce the product launching time is imposed on the participants [2].

One of the generic challenges small organizations face is to reduce the timeline from product requirements to deployment [3]. Third party dependency is one of the recurring themes in software startups due to their lack of resources, time pressure, small and low-experienced team [4]. An organization which is in its initial phase of operation and dependent on its software platform to deliver the product to its customer is quite unlikely to develop the software within the organization, rather it engages itself into product requirements, business process development which are implemented in the product, optimization of overall product ecosystem, pricing, marketing, promotion and sales. Shifting most of the technologically risky components of the value delivery chain to some other parties have always been a wise option for startups.

However such organizations cannot avoid the risk of being too much dependent on the partner as the company grows [24]. As the product grows in many directions, the detail knowledge of the product keeps growing beyond the boundary of an organization's competency level. One possible option to reduce the risk of being dependent on a single partner could be to engage multiple partners in different components of the product [25]. That could bring manageability and compatibility issues among the partners [26]. Another option could be to acquire the source code of critical components, enhance organization's

knowledge base and involve in small scale development from where it can grow further. The route an organization prefers to select depends on the strategic goals of the organization. But it is a common problem for any similar organization in their initial phases. A defined delivery pipeline starting from the product requirement to release could be an essential starting point on which an organization can plan the requirement of tools, automation, process engineering or infrastructural adjustments.

In this master thesis I attempt to study a software startup, which is delivering digital services to its customer and dependent on its partners to provide those services, analyze its product delivery value chain and attempt to define the delivery pipeline, which could be used as the baseline document from where necessary actions to achieve the strategic objectives such as reducing the timeline for releasing new features in the product, and mitigate the risks of depending on a single partner could be identified.

1.2 Justification of this study

According to my literature and background study in databases such as IEEEExplore, ScienceDirect, ACM, the term delivery pipeline is used in different software engineering approaches such as Continuous Integration (CI), Continuous Delivery (CDE) and Continuous Deployment (CD). In these scenarios the delivery process starts from committing the source code into the repository and ends at different stages in the delivery pipeline [17]. An end to end process description starting from the requirement study to the release is missing. It could be due to the fact that, there are variations in the approaches taken by the organization considering specific cases. However, my study is specific to the organization which already has an approach for software development, and such an approach could be generalized for similar kind of startups in the industry.

Startups are mostly engaged on managing the most critical processes for achieving its business goals and expand thereafter vertically or horizontally [27]. Involvement into the

detailed execution of the processes, which are not yet directly explored by the organization are likely to be outsourced to experts to achieve several business benefits.

A startup does not have enough human resources to manage every aspect of the business with enough expert knowledge [10]. Role playing is one of the characteristics in startups. There are several things to manage when the product is already in production. As it is a new product, if it does not have enough features which could be perceived to add value to the customer, and is not quality tested then it might have enough risk of being rejected. A newly launched product rejected by the customers for the first time would require tremendous effort to revive its position in market. If the product is rejected by the end users, the pharmacists will lose their interests to subscribe to the service. The product has to incorporate all highly valued functionalities, non-functional features for increasing usability, and legal requirements set by authorities for compliance and needs to be released after testing the quality as soon as possible. Hence the organization need to manage challenges such as resource scarcity, customer satisfaction, legal compliance, quality, usability and go to market. So an automated delivery pipeline would be required more than anything else in the first place.

1.3 Goals and delimitations

The goal of this master thesis is to propose a delivery pipeline for the organization, which would be backed up by the industry best practices for a small startup organization. Hence the relevant practices are studied, tools are identified, the product is analyzed, required tools are implemented in a simulation setup and the outcome is analyzed along with the business objectives.

I attempt to answer the following two higher level research questions in this study,

RQ1: What could be an ideal delivery pipeline for PharmaServe?

RQ2: What could be the challenges in setting up a delivery pipeline for PharmaServe?

1.4 Structure of the thesis

In section 2, I introduce the relevant terms which are to be used in the remaining of the thesis

In section 3, I describe the organization, its product, the business objectives of the organization, the generalized business objectives for similar kind of companies.

I describe the research questions and the research methodology I have followed to answer those questions.

In section 3, I attempt to do the groundwork for deriving the delivery pipeline through the identification of the objectives of the solution.

In section 4, I describe the recommendations for the organization for an optimum delivery pipeline.

In section 5, the delivery pipeline and its different components are described

Finally, in section 6, challenges of implementing the delivery pipeline, improvement areas and future directions are discussed.

2 LITERATURE ON DELIVERY PIPELINES

2.1 Relevant Literature

It is important to have a summarized idea of the concept described till now to identify the relevant literature in the subject. The summarized idea is to have “An automated delivery pipeline for a small startup organization, which is dependent on a development partner for the development of the core product”. Analyzing the summarized concept indicates that, existing literature need to be studied to understand the characteristics of startup organizations, delivery pipeline, continuous practices, software development outsourcing and business process management.

2.1.1 Startup Organization

A business is an entity involved in the provision of goods and/or services to consumers, and software business refers to the commercial activity of the software industry, aimed at generating income from delivery of software products and software services. It shares some common features with other knowledge-intensive businesses [6]. Startups are a special form of businesses having different set of characteristics due to their length of time in the market. There are many definitions of startup companies in the academic study areas. It is an organization which is temporary in nature, exploring the opportunity of a scalable, repeatable and profitable business model [8]. Sometimes startups are defined as human institutions aimed to create new products or services in extreme uncertain conditions [9]. It is also defined as organizations with significantly less experience of operating in the market, having inadequate resources and influence by several organizational factors such as investors, customers, competitors, and the use of dynamic technologies [10]. According to [11] there are certain principles and practices in the startup such as

- Process management is Agile, Evolutionary and Opportunistic
- Software development is driven by customers who act as designers
- The team is the catalyst of the development

- Tools can accommodate product and management changes

In this paper [10] the author has summarized some common reasons why software startup fails, which are

- Inexperienced developers
- Excess customization over cored product development
- Lack of product ownership
- Absence of strategic plan for the product development
- Unrecognized product platform

To summarize the challenges of a startup company, we can say that a startup has

- Lack of skilled and experienced resources for product development
- The time to market pressure is high due to fierce competition in the market
- Financial pressures from the investors
- uncertainty of not accepting the product by the market
- deviation from the core business focus, and
- dependency on third parties for product or services development

Software Business Model is the configurations of the firm's offering, activities, value network, and revenue logic, given its competitive strategy, resources, and competitive environment [7].

2.1.2 Delivery Pipeline

The objective of any business organization is to provide some service or product to its customers, and a startup aimed to deliver some innovative services to customers need to organize the product development by means of engaging internal resources or external parties. The ultimate objective of the organization is to produce some value to the customers [50]. Hence the value creation starts with product requirements, development, testing and quality assurance and finally deployment so that the service could be used by the intended customers. The entire process of creating the product or service including all

details subprocesses could be termed as the delivery pipeline for the organization. It is important for an organization, which is engaged in IT or digital services to focus on requirement specification, design and development of the product, testing, and release. In the requirement specification, the functional and non-functional features are identified, evaluated. In the design and development phase a product prototype is designed with the features identified in the requirement phase. In the testing phase the product is tested from different perspectives to see if the product delivers its intended outcome. In the deployment phase the product is released and made available for the customers, so that customers can use it.

Due to the several limitations of a startup organization as mentioned earlier, and to focus on the core business processes it is quite common to depend on third parties for non-core operations. It could be a common pattern for startup organization which intends to provide services through a software platform, and requires some software product through which the services are delivered and the core focus of the organization is not to evolve as a software development company then to outsource its software design and development part and engage itself with product design, testing, release and other operational activities.

2.1.3 Continuous Practices

There are several practices in the software development areas such as continuous integration, continuous delivery and continuous deployment, which are termed as continuous practices all together.

Continuous Integration (CI) is an agile software development practice that emphasizes team members in a software development team to check in code into a shared repository frequently so that each check in is verified by an automated build, enabling team members to detect the problems and resolve accordingly [28]. The objective of continuous integration is to ensure a work in progress product workable [29], stable although limited in features. While software engineers are working in developing some product features, they are encouraged to check in the source code to the server without breaking the stability

of the product, by executing an automated script. While the source code is checked in, the automated script executes, which includes several automated test cases. If the test cases are passed then it is assumed that the commit did not break the product. If the tests are not passed the developer is notified so that the committed could be taken out and work on that and check in again. With the wide acceptance of agile software development methods, the use CI practice has increased in many software development organizations due to its inherent benefits such as rapid deployment. Gartner identified the foundational practices of rapid deployment as one of the top ten strategic technology trends for 2015. [12] In a recent survey while analyzing the practice to what extent continuous delivery are used among the 19 software development organization in USA, it was found that automated testing and automated deployment ranked in top two among 11 practices. [13]

Continuous delivery is also a software engineering practice, where the code commit by an engineer triggers automated test execution, and building the product so that the feature could be tested by the users. It includes some processes in addition to continuous integration. [30][31][32]

Continuous deployment includes some more extra processes in addition to continuous delivery. It means that the product will be automatically ready for deployment after the codes are committed by the engineers. [31][33][34]

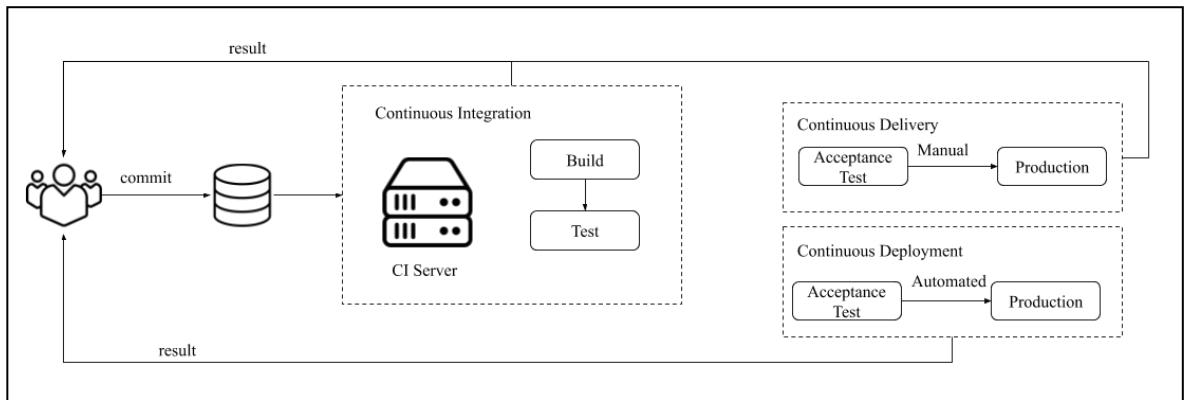


Figure - 1: *Continuous Integration, Continuous Delivery and Continuous Deployment,*
redrawn from [17]

2.1.4 Software Development Outsourcing (SDO) [35][36][37][38]

There are four major types of sourcing options for organization such as, offshore outsourcing, onshore outsourcing, offshore insourcing and onshore insourcing [14]. Outsourcing refers to getting the things done by some other organization according to the agreed contract by both the parties. Offshore outsourcing refers to contracting some organization which operates in a different country. The organization to which this study is related is currently engaged in onshore outsourcing. Most of the studies regarding outsourcing are based on large organization, information on small and medium enterprises are quite less.

2.1.5 Business Process Management (BPM)

A business process could be defined as a group of interrelated events, activities and decisions based on several factors, which is always triggered by a business need and intended to produce an outcome for the customers. Business Process Management is the body of principles, methods and tools to design, analyze, execute and monitor business processes with the aim of improving their performances [15].

3 RESEARCH ENVIRONMENT AND METHODOLOGY

3.1 The organization and the product

For the purpose of this study I consider a typical small software startup (for instance PharmaServe), which has less than 10 employees and founded within the last two years. One of the long term strategic objectives for the PharmaServe is to evolve as customer service and delivery solution provider for the health service providers by means of state of the art technologies. Currently it provides customer service solutions to the pharmacies in Finland, through which pharmacies can provide remote customer services to their customers.

The service is provided through a cloud based service platform namely VideoCon. VideoCon used by the pharmacies to provide video services, is a web application and can be connected either from mobile applications or applications installed on kiosks. Pharmacy customers need to install mobile applications on their mobile device to get the service, or need to visit a nearby kiosk. Kiosk are set up at different locations by the PharmaServe, where the client applications are installed, through which users can receive the service from pharmacies. Apps installed on handheld mobile devices or kiosk applications display a list of pharmacies which has registered for the VideoCon services. Pharmacists at the pharmacies log in to the VideoCon system and make them available so that customers can see the available pharmacies, select a pharmacy from the list, call, enter either into video or audio call and order pharmacy products. The ordered products can either be picked up from pharmacy, or picked up from pickup stations or delivered to home according to the preference of the customer and the capabilities of the pharmacies. To supplement the customer service solutions PharmaServe has another product which includes delivery boxes at different locations. Pharmacies need to subscribe to those delivery boxes. If a customer orders pharmacy products and prefers to collect the product from any delivery box, the product is delivered to the delivery box and the customer is informed through text messages.

There are six major components in the product.

Web Portal: The web portal is the primary interface, which gives a brief introduction to the system for the users. It also acts as the interface through which the potential customers or the pharmacies express their intention to subscribe to the system. The order details are sent as an email to the person in the organization responsible for managing the orders. The web portal has all the user manuals for the system. The subscribed pharmacies have access to the user manuals.

Super Admin Module: The super admin module is operated by the PharmaServe to manage system wide configurations such as pharmacies, users, kiosks, other application wide configuration information, data usage, payments, billing.

PharmacyApp: Application used by the pharmacies. The major features included in the PharmacyApp are, pharmacy data configuration, user management, customer service, opening or closing the queue. To use the system, some basic pharmacy information such as name, address, opening hours, supported delivery methods need to be configured first. Users are added to the system and can have either admin or pharmacist privilege. Admin has access to pharmacy configuration system in addition to regular pharmacist privilege. The default and the basic screen of the system is to receive call, enter into video conference, provide service, end call. When a user logs in his/her availability status is online, the pharmacy will appear available to receive call in the mobile apps and kiosk apps. The default screen appears, which shows the number of customers in the queue at this moment. Users from the mobile app or kiosk app can initiate call to the pharmacy. The queue management feature enables pharmacies to handle the calls in a first come first serve basis. Pharmacist can change their availability status to away, and if all the pharmacists in a pharmacy have their statuses away, the pharmacy will appear offline in the mobile or kiosk. Pharmacist can take a call, enter into video conference and provide service to the customer. Another important feature is service language. A pharmacy can have many pharmacists who are able to provide service in different languages. If the service language of a pharmacist matches with that of the preference set by a user in his/her mobile app, that

pharmacy will be shown available to receive call. The default language for a pharmacy is Finnish. The call logs and order details are also preserved in the system so that pharmacies can process the order after the call is ended, and the product is delivered according to the agreement with the user. Computers used in the pharmacies with standard internet connection, a web camera and a headset are basic infrastructural requirements to use PharmacyApp from a pharmacy.

MobileApp: There are currently two versions of this app, such as android and ios. Customers need to download the app from play store or app store and install on their mobile devices. The users need to register to the system, provide basic information such as address, preferred service language, payment card information. The default screen of the app is the list of pharmacies sorted on their distance from the caller, availability. A user can mark one or more pharmacies as favorites, sort the list according to distance from home address or current address. A user selects a pharmacy and initiates a call to that pharmacy. The caller is added to the pharmacy queue, when the call is received by a pharmacist the caller and the pharmacies enter into a video conference call and avail the service as a walk in customer. While the pharmacist at the pharmacy end is serving another customer, the calling customer will be kept in the queue. When the pharmacist receives the call, a call back alarm is played on the caller's phone to receive the call and enter into the conference.

KioskApp: These are additional interfaces to the system at some fixed locations, from where customers can contact with the pharmacies. A kiosk has an android touch panel, a headset and two web cameras attached to it. The application always shows the pharmacies, and the number of customers in the queue of that pharmacy. A pharmacy to be displayed on a kiosk requires to subscribe kiosk modules separately. Customer picks the headset and presses the icon of the pharmacy, he/she wishes to call, the call enters into the queue, and if any pharmacist receives the call, the caller and the pharmacist enter into video conference. The secondary webcam acts as the card reader to which the customer shows any valid identity card. Currently kiosk does not have any payment option.

Queue Indicator: It is an optional component some pharmacies can use if they require to do so. It is a small application running on an android tab, which shows the queue count of a pharmacy. The purpose of this component is to draw the attention of the pharmacists if there are customers waiting in the queue. As most of the time pharmacists are busy with their walk in patient, and it is not always practical to log in to the system to check if there are online customers, hence this tab is placed in some place from where anybody can see, and whenever there is a customer joins the queue an alarm rings.

The revenue is generated from the initial setup fee from the VideoCon, monthly subscription fee for the VideoCon and delivery boxes, and the video data usage.

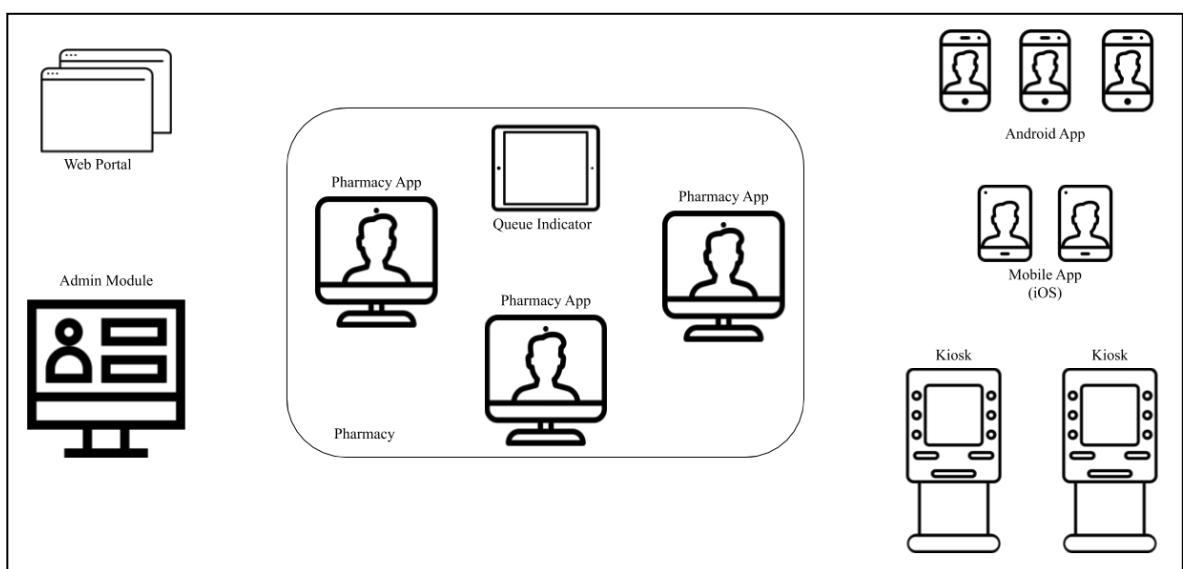


Figure - 2: *VideoCon System Components*

SysDev is a development partner for the PharmaServe, which develops the application and different supplementary modules according to the requirements provided by PharmaServe. At PharmaServe, the application is tested, verified and validated according to the functional and quality requirements, and then published to the relevant distribution channels such as google cloud server, android play store and apple app store.

3.2 The development process

PharmaServe maintains a product catalog which lists the product features either implemented or not yet implemented in the product. Informal descriptions of the features are managed in the catalog. Product catalog is the output of the requirements, gathered in different ways such as, ideas given by the executives, feedback from the current user group, regulatory compliance. The requirements are mainly prioritized according to their values delivered such as functionality, usability, and operability.

The software delivery process starts with the brief requirements specification document at PharmaServe. A brief document is shared with the development partner and discussed for further clarification. If there is any user interface related work then the development partner develops the mock ups based on the document and subsequent discussions. The mockups are shared with PharmaServe for their inputs and agreement. The development partner maintains an online kanban board, where the progress of the items tracked in different stages, and the subscribers to that kanban board is notified if an item has moved to any column. The software components are broken down into to-do lists and entered a kanban board to monitor their progress in the development. There are mainly six columns in the kanban board, which are backlog, to-do, in-progress, done, testing done, and release. The items in the backlog column are entered by the PharmaServe, which are the outputs of generated and prioritized requirements from different sources. These are entered into backlog column so that the developers can get acquainted with the upcoming features and their technological readiness if required. A weekly meeting between the PharmaServe and SysDev is held where different ongoing issues are discussed, one of the important issues is the upcoming items to be developed in the next week. Some of the items from the backlog are agreed on based on their perceived value to be produced, technological readiness, availability of resources and developers to work in the next cycle. Such items are moved from backlog to to-do column. So the items in the to-do list are assigned by the PharmaServe to SysDev, of which resource engagement are tracked and paid afterwards. When an item from the todo list is picked by a developer he/she moves that to the

in-progress column. The developer works on that item in his/her local development environment. Once that item is completed the item is moved to done column. An item done in the kanban board means that, the developer has completed the task in his local environment but yet is not available for someone to test. After the deliverable or the necessary code segments are committed to testing server, the item is moved to ready for testing column. At this point the testers in the PharmaServe are notified and start testing the deliverable. If an item passes the testing then it is moved to testing done column. If there are a significant number of items in the testing done column, the developers along with the pharmaserve decide to make a release. The release is made but not deployed on the production until final decision. There are mainly three distribution channel for the VideoCon. The web platform which is used by pharmacists is hosted on google cloud platform. The android app is hosted on play store and ios app is hosted on the app store. The pharmacies are informed about the changes, or newly added feature by mail. The user manuals are updated accordingly.

Stakeholders of the PharmaServe: According to the current product and service offerings there are four major stakeholder groups, which are directly involved with the delivery pipeline.

Pharmacies: Pharmacies are the primary user of the system and main customer. Pharmacy owner decides to obtain the service and install the system at their pharmacies. There are two kinds of uses in the pharmacy. Admin user can configure the address, delivery methods, opening hours, close the queue and manage other users in the pharmacy in addition to regular customer service.

Pharmacy Customers: Pharmacy customers are the end users of the system, who receive the service by installing mobile apps on their handheld devices or by visiting the nearest kiosk. This group of stakeholders is quite important as the usability features of the system are mostly derived considering the needs of this group. The decisions regarding the installation of the kiosks and delivery boxes are also impacted by pharmacy customers.

Development partner: SysDev is the primary development partner for PharmaServe. Currently resources from SysDev are contracted at an hourly rate. The tasks are agreed earlier and the efforts are measured by SysDev and reported monthly to PharmaServe for payments.

Marketing: Persons responsible for marketing and promotion required to be informed and trained properly so that can identify the unique sales proposition and utilize such for different promotional activities. Quite often marketing has inputs to the product features.

3.3 Current delivery pipeline

Considering the current development and operational processes of the organization, the delivery pipeline for the PharmaServe can be considered from the processes starting from product requirements collection in the product backlog and all the way to release notes communication to relevant stakeholders.

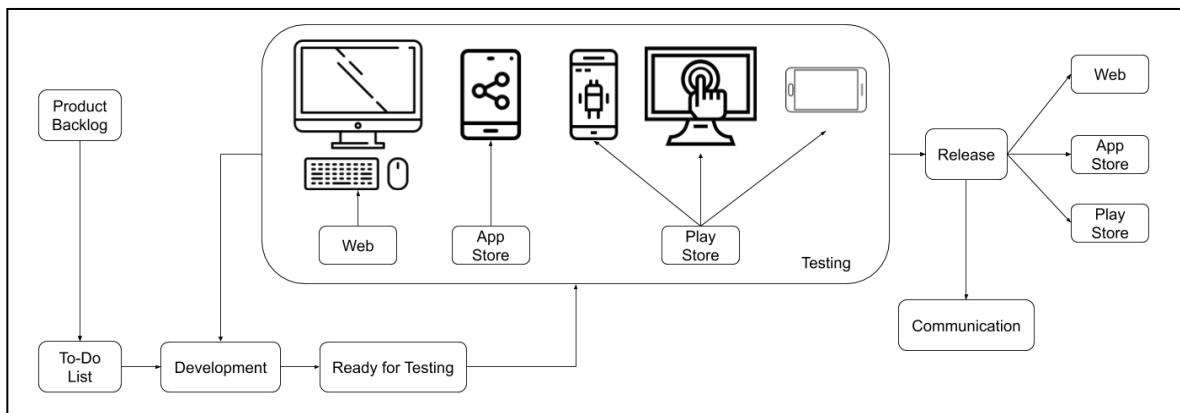


Figure - 3: *Current Delivery Pipeline*

3.4 Research Methodology

The purpose of this study is to define the pipeline, find out the opportunities where the pipeline could be enhanced in a way that could have positive contributions towards

achieving the strategic goals of the organization. A defined and optimized delivery pipeline based on this research is assumed to contribute to the following higher level objectives

- provide a guideline for setting up a delivery pipeline for a typical small startup engaged in delivering digital services
- Identify the possible enhancement points and strive for continuous improvement

I attempt to answer the following two higher level research questions in this study,

RQ1: What could be an ideal delivery pipeline for PharmaServe?

RQ2: What could be the challenges in setting up a delivery pipeline for PharmaServe?

This is a design science research as I am going to build an artifact based on the business needs of the organization and evaluate the same. The higher level methodology to be followed to come up with an ideal delivery pipeline is to follow a formal design science research method. Hence the DSRM [5] process model has been used. There are six processes in the design science research method for software systems and each process is targeted to produce an output following a specific activity. Out of these six processes, I followed three, which include define objectives of a solution, design & development, and the demonstration. In the definition of the objectives and the solution process, I define what higher level objectives I target to achieve and what solution I am expecting to produce out of this research. In the design and development process, I come up with the solution based on other identified relevant components. And finally in the demonstration process I explain the solution and demonstrate how it could be used in the organization and how it could produce the expected goals. I exclude three other processes in the DSRM, which are problem identification and motivation, evaluation and communication. In the introduction section of this report, the problem identification and the motivation is explained to some extent. The evaluation of the solution is supposed to be done in the real life situation after using the solution for a significant period, hence considering the scope and time constraint of this study evaluation and communication steps are excluded.

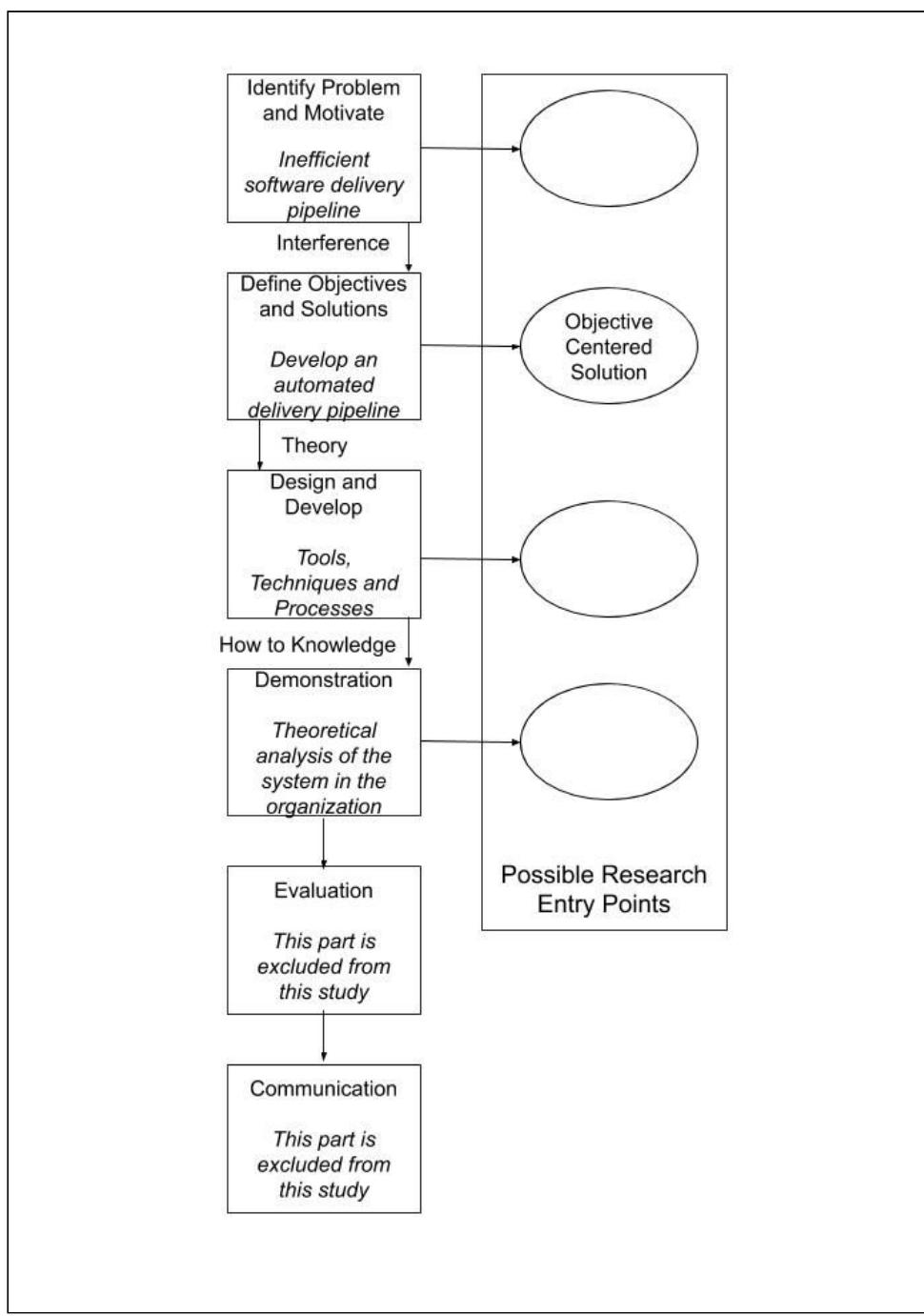


Figure 4. DSRM Process for the Study. Redrawn from [5]

Like software architecture in a software product, each and every organization operating in the market has a product or service delivery pipeline, even though it is not explicitly defined or formally accepted by the organization. To answer the higher level research questions it is important to understand the environment, what different organizational units are involved in the entire process through which the product or the service is being delivered to customers. It is also important to understand the technological infrastructure through which the service is being delivered. The development process starting from the product requirement analysis to product release, maintenance and support are also important to identify the current delivery pipeline. Existing product documentation and organizational knowledge base are the only source which could be utilized to gather relevant information, hence documentation and available literature reviews will be used for answering these questions.

Relevant literature from different research databases such as IEEEExplore, ScienceDirect, ACM, Researchgate will be used to study the relevant literature to find out the industry practices of delivery pipeline. A limited literature review on outsourcing software development has been conducted, the outcome of which could also be utilized as the software development of the organization is outsourced to some domestic partner, and outsourcing is one of the key aspects of small startups. The literature would give a summary what are the outsourcing model, practices and what challenges are there to consider in outsourcing the development.

Due to the fact that the current study does not have any end to end process for a small startup, I need to organize the results of studies done in fragmented way to relate to the case. Some of the relevant studies I have identified are Software Process Improvement, Lean Software Development, Continuous Practices, Global Software Development, Software Development Outsourcing, Business Process Management, IT Service Management, Startup characteristics to name a few.

A systematic theoretical analysis for evaluating the to-be delivery pipeline would be conducted against with the organization's strategic business objectives to find out the challenges it requires to consider.

It should be taken into consideration that, most of the studies found in the literature review are not directly for small startups, hence I would like to utilize the learning which could be generic for any organization delivering digital services.

4 OBJECTIVES OF THE SOLUTION

The outcome of this study is an artifact for the PharmaServe which can be used to address the challenges identified in the previous chapter. The artifact is a delivery pipeline or the process flow diagram, which includes all the major processes starting from the requirements to release. The diagram will highlight the critical checkpoints in the pipeline, identify the enhancement opportunities with the use of tools, techniques or process reengineering.

The to-be delivery pipeline is expected to reduce requirement to release timeline by identifying critical checkpoints, recommending tools for automation in different subprocesses. The pipeline will also highlight how to initiate the process of acquiring knowledge of the developed components, develop, own the repository and ensure the stability and proper reflection of the product growth with upcoming releases. Currently the organization is not involved in software development, because the core competency is to optimize the customer service solutions for the pharmacies. If there is a defined agreed process of entire value delivery chain, the organization can focus on the product features triggered by the market, regulatory and internationalization. It will also show an opportunity how multiple partners could be engaged in the development.

Objectives	How
Reduce requirement to release timeline	Tools
Code ownership	Tools and Processes
Engage multiple partners	Tools and Processes

Table - 1: *Objectives of the Solution*

Currently the organization does not have any recorded data regarding the usual time taken from a new requirement to release. Requirements in the product catalog maintained by the

organization are recorded informally and frequent informal meetings are held to identify the features to be implemented in the system. Such meetings are mainly triggered by the issues identified during support to some customers. However a requirement imposed by the regulatory authority which was not known earlier in the product development phase and product enhancement requirements given by the customers after using the system, are always given higher priority. In some cases the requirements are not even recorded in the product catalog, rather discussed with the developers regarding the technical feasibility of the solution and availability of resources to accomplish the development, and afterwards the task is directly entered into the kanban to-do list. Informal discussions are held in case some clarifications are required. So no formal record of time required to convert an identified requirement to a product feature ready to be used by the customer is unknown. With the newly developed delivery pipeline, the output could be evaluated.

The produced delivery pipeline could be used as a benchmark process for delivering the value to customers, and can be analyzed for further optimization. It also specifies the processes involved in converting the requirements into product, hence it could be an initial artifact based on which further software process improvement initiatives could be undertaken.

In addition to the objectives mentioned above, the artifact needs to be feasible to use for the organization. There are several tools, techniques and process reengineering, which requires to be usable by the organization with such limited resources. Introduction of this artifact should not make the current business process more complex, incur additional financial expenses.

5 RECOMMENDED DELIVERY PIPELINE

The objective of this chapter is to identify the tools, techniques and processes for the potential delivery pipeline. The reasoning behind selecting certain tools, techniques, and processes will be explained.

First of all I identify the core operations in the technical areas of the organization and their major activities. Then the activities are evaluated to see if and how these activities are connected with the delivery pipeline. All the activities which are directly connected with the delivery pipeline are grouped according to their similarity in different concerns. A theoretical analysis based on literature are performed among these concerns, and objectives identified in the previous chapter.

The key objective of the technical department of PharmaServe is to manage all technology related aspects, which are essential for PharmaServe to operate in the market with the product and service offerings. Conceptually there are six units of operations identified in different discussions in the technical department, which are

- Product management
- Software design and development
- Quality Assurance & Testing
- Release Management
- Installation, Training, Support and Maintenance
- Infrastructure Management

Product Management: The primary purpose of the product management unit is to maintain the product backlog, which keeps all the features of the product are either already developed or yet to be developed and the releases. All necessary activities to identify, capture, and document the product requirements triggered by customer needs, market demand, technological advancement, regulatory requirements. It also includes the

evaluation of the product requirements by following necessary prioritization techniques. Prioritization may take input from architecture to measure the change.

Software Design and Development: Although the organization is not currently engaged in software development directly, however one of its targets is to acquire the software development activities of the critical components in house to avoid the risk of dependency on a single partner. The initial objective is to setup the development environment and contribute in the development by adding small features. The purpose of this unit is to manage the transformation of the product requirements into product features, by following proper development route. The acceptance criteria for the changes are documented, which are the base documents for acceptance testing. Software development also include the identification, capture, enhance, and documenting the product architecture. Design business processes which are essential for the product.

Testing & Quality Assurance: The purpose of the Testing and Quality Assurance unit is to ensure that the service delivered to customers with the products are ensured to meet certain quality parameters. The quality targets are defined and to meet those quality targets testings are conducted. Tasks in this unit include, set the quality target, identify the quality parameters, manage the testing and quality assurance processes to ensure the product achieves the quality targets.

Release Management: There are seven different components in the system and six different release environment. The purpose of this unit is to ensure the quality checked product is available for its intended customers to use, and make aware the customers with the with proper communication materials.

Installation, Training, Support and Maintenance: This unit is responsible for installing the system at client site, train them so that they can use the system and handle the support issues. It is also important to manage the inventory of the peripherals such as web cameras, mouth speaker, record the installation logs for smooth support and maintenance. Keep the

installation manual alive. Develop training materials, conduct training, document training conducted, customer feedback. Follow up maintenance requirements for the customers, manage the support channels, resolve issues, escalate to necessary channels if required.

Infrastructure Management: This unit is responsible for managing all infrastructural requirements for the organization. It requires to manage the google cloud services, video service providers, hosting services, JIRA and Confluence service.

After analyzing the current delivery pipeline and the major activities in the units the following initial improvement opportunities have been identified,

- Enhance the software configuration management processes
- Enhance Testing and Quality Assurance Processes
- Introduce Continuous Practices

Enhance the Software Configuration Management Processes

A software system is a combination of several components interacting together to provide a service as a whole to its intended users. Configuration means the specific versions of the participating elements in the system [39]. The purpose of software configuration management (SCM) is to identify the elements, manage the change controls, ensuring the integrity and traceability throughout the system development life cycle. The evolving software system needs to be tracked throughout the development life cycle starting from the requirement management to release, otherwise it is prone to forget what changes have been incorporated in which version of a software component. Although version control is one of the most visible activities of software configuration management, there are four major activities in software configuration management, such as version control, system building, change management, and release management [39].

Version Control: It involves the management of the different versions of a software component. Some version numbering systems are used to identify different versions.

System Building: It involves compiling the components and making the system ready so that it can be tested or used in some limited environment.

Change Management: It involves the activities which are required to incorporate some changes in the system. An appropriate change control system can help the organization in many ways such as, it can avoid incorporating features which are not needed, control on the cost in using resources for making the change.

Release Management: It involves all the activities making the system ready for its intended users, making it available in the appropriate distribution channel and informing the customers regarding the changes. It also keeps track of the version numbers of the components.

Configuration management has been identified one of the enhancement areas for the PharmaServe due to three basic reasons. Firstly, this particular process takes place in the value delivery pipeline, enhancing these processes has a direct impact on reducing the timeline of the system delivery. Secondly, the system development is outsourced to the development partner, and these activities are mostly neglected as these are not given proper attention by the business people. The software product is visible to the management through their features and usage pattern by the customers, however, configuration activities are to help the organization in the long run in software maintenance. Finally, software configuration management is directly connected with the software quality assurance [40]. Configuration management activities help organization to achieve its software quality assurance goals by providing adequate confidence that qualities are built in within the system.

The following practices from the configuration management can be introduced in the organization.

Recommendation-1: Bitbucket repository: Currently the development partner uses their own repository to manage the source code of their projects. Source code of the VideoCon systems are checked in to the repository according to the project management methodology followed by the partner company. When one or more features are developed and the product is ready to demonstrate the output, then the product is built and deployed on the testing environment for the PharmaServe to test and validate the features. PharmaServe could create their own repository and request the developers to check in the code from their local environment to the central repository.

PharmaServe currently uses Jira [41] and Confluence [42] in some limited scale to manage its documentation. Bitbucket [43] is one of the web based version control systems from the same provider Atlassian. It also gives an easy integration with the Jira and Confluence system [44]. Bitbucket or similar repositories enable to create multiple branches of the main line, on which developer can work and check in their works without disturbing the stability of the main line. By taking control of the source code, PharmaServe can incorporate other developers into the development.

Recommendation - 2: Automated system building process: There are currently seven different components as described earlier in the VideoCon system. These seven components can be categorized into four groups according to the underlying technologies used for development. The system has been developed using different components mostly from Ionic [45] and Firebase [46]. Ionic is JavaScript based open source hybrid application development framework. Applications developed in Ionic framework can be compiled according to specific target platform requirements such as iOS or android. Firebase is the Google's Backend As A Service (BAAS) on which mobile and web applications can be developed. Different components such as authentication, realtime database, storage, cloud functions were used to develop the backends. One of the reasons for selecting Ionic

Firebase platform is to reduce the development time by avoiding platform specific component development. The web applications such as Pharmacy Web, Super Admin and the mobile applications for Android and iOS are developed in Ionic - Firebase and Ionic - Cordova [47] - Firebase respectively. Twilio [48] video service has been used for video conferencing system. Twilio is a communication platform which enables applications to have different communication features such as phone calls, conference calls, video conferences. Due to the requirement of android native libraries for the Kiosk application, the Ionic, Cordova were discarded for this specific component. rather android SDK (Software Development Kit) has been used. The decision was taken after several weeks of experiments, where it was revealed that to use two video cameras and a mouth speaker which are connected to the android touch panel through USB ports, the native libraries of the android SDK need to be used. Hence a different development technology has been chosen. The web site of the organization, which serves as the primary interface and instruction manuals for the system is developed using Wordpress. Wordpress is a PHP MySQL based content management system. If a new feature is added to any other components in the system, the instruction manuals are updated in the website.

If the information update in the website could be discarded from system building due to the insignificant amount of work, then there are still four different build processes need to be managed for the system to be deployed on the testing system. Hence these could be automated to reduce the time. The build scripts could be developed, which collects all relevant components according to the configuration files and build the system along with all required libraries, deploy the system for testing and notify the users of the testing system.

Recommendation - 3: Automated Release Management: There are currently seven different deployment environments to which release processes need to be managed, for the VideoCon system. Any change in the business logic is accomplished by changing in the cloud functions [49], hence the cloud functions need to be released to the google cloud environment so that other components such as pharmacy web, super admin mobile apps

can use the underlying updated business logic. Changes in the user interface of the Super Admin or the Pharmacy Admin trigger release new versions in the firebase hosting environment. Changes in the user interface or in the features of mobile application trigger release new versions in the Play Store and App Store so that these can be used by the end users. The kiosk application is not meant for the end users to install on any device. There is kiosk management system in the PharmaServe which takes care of all the kiosk and their installed applications remotely. The built apk file need to be uploaded into the system and a remote deployment process is run, which updates the kiosks with the latest application. The Queue Indicator is delivered to the clients pre-installed, hence the built apk file is manually transferred to the tablet and updated. The reason it is not released in the play store is, the application should not be available for the mass people. And the website is released but it does not have any impact on the user base. Currently after PharmaServe validates the features by performing testing and quality procedures, it informs the development partner, who run the release scripts and releases to different environments. The release management processes could be automated so that when the testing is done by the PharmaServe, the release scripts could execute with some minimum effort from PharmaServe and they can decide when to release.

Enhance Testing and Quality Assurance Processes

The primary purpose of software testing is to demonstrate that the developed component produces expected results and to discover the defects in the component so that those can be resolved before putting it into actual use [39]. There are several benefits of testing and assuring the quality of the product such as, first of all, it reduced the cost of maintenance and support. Secondly, it increases the reliability on the product. Being an innovative product in the market and to motivate the users to use the system, the product requires to be error free and easy to use. If the product is not stable there is a huge possibility of being rejected by the users. In this competitive market, the user will switch to another substitute of the product. And customer acquisition is always more expensive than retention.

According to SWEBOK [40], software testing can be categorized based on two different perspectives such as the target and the objective. According to target of the test, software testing can be categorized in three different stages such as unit testing, integration testing and system testing. Unit testing verifies if a component at its minute level functions properly, and the test cases are developed by the developers of the component. The test cases uses the software components with test data to see if the expected results are produced when the test case is executed. Although the unit test does not ensure the correctness of the functionality but the ultimate purpose is to maintain stability in the codebase, or to see if the code works in the same way after making some changes. Integration testing verifies how components work together to produce some functionality. There are mainly two approaches for integration testing such as bottom-up approach or top-down approach. For small systems often Big-Bang testing strategies are followed. System testing is done to check the behavior of the entire system as a whole. Quite often system testing is not performed by the developers. Based on the objectives of testing there are thirteen types as mentioned in the SWEBOK [40]. Acceptance testing is done to check if the system requirements are properly implemented in the developed system and they are acceptable by the customers. Installation testing refers to the testing process where the system is installed on the target environment to see if it works properly. Alpha testing refers to exposing the system to a small group within the organization to check if the system works properly before going to further in the release process. Beta testing refers to exposing the system to small group of users who are not necessarily part of the organization. With the help of Google Console and App Store by Android and Apple respectively the alpha and beta testing have become quite simple to execute in terms of infrastructure and organization. Regression testing refers to the testing process which intends to check if the underlying system functionalities have not been affected due to the introduction of any change. Performance testing refers to check if the system performs according to the expectation. Such as end to end call establishment time is expected to be less than five seconds if the pharmacist is available to take a call. Security testing involves the verification process to check if the system is free from any unwanted threats from intruders. There are certain security concerns in the system such as, only the valid and

actual users should be able to provide their consent to pharmacists so that they can access their health and medical information. The delivery information and pin-code (if required) should only be sent to the customers of the medicine. Stress testing refer to check how much load the system can take before it starts to behave abnormally. This information is very important to be communicated to the customers, so that they can enhance their capacity to support additional load. Recovery testing refers to check if the system recovers to normal state after some abnormal behavior. The system is highly dependent on different third party components such as internet speed, twilio API so if there is any exceptional exit during any process execution, the system should come back to the original state so that normal operation could be continued. Usability testing refers to evaluating system how much user friendly the system is. One of the biggest chunks of users are from old user group for whom a new system should be quite easy to use. Different system features are added to make a system more usable, such as height adjustable desk for the kiosk users, call back alarm in case a calling user is kept in the queue and subsequently the call is received by a pharmacist.

The testing and quality assurance unit in the PharmaServe is responsible for all sort of testing and quality assurance. The development partner is responsible for development and releasing the product into the testing environment, from where the testing process starts. Due to the limitation in resources and go to market pressure the functionalities are checked manually by executing the most important features on target environment. The unit testing and integration testing are not done at the development end.

Recommendation - 4: Set up a complete testing lab with proper infrastructure

Currently the most important thing to focus for PharmaServe is, the testing and quality assurance processes. A proper infrastructure with required testing devices and environment could contribute tremendously to the product quality. Some testing devices such as a laptop with windows operating system, two android phones, two iPhones, two android tabs and two android based touch panels could be a good investment. The devices could be marked

and identifications used in these devices could be used to create an internal testing group so that the test version of the software is directly distributed to these devices.

Recommendation - 5: Use testing tools and introduce automated testing

Test automation is one of the key prerequisites for implementing automated delivery pipeline [53]. Automated testing could benefit PharmaServe in many ways. Firstly, it could increase the test coverage of all the products. A selective manual testing is prone to discard some critical features, which are essential for the system to run. Secondly, it could reduce the level of effort in the testing process. Currently it takes five to seven working days to test the whole system with some selected features, however in automated testing time could be reduced significantly. Thirdly, the resource could be engaged in more productive things which will contribute to the long term objectives of the organization. Persons involved in testing could be engaged in writing test cases and executing those test cases. By writing test cases the knowledge of the system is acquired by the organization, which could help in mitigating risk of dependency on a single development partner.

A few most critical use cases could be selected for automated testing initially. API based testing of the selected use cases could be started considering the convenience of initiating the automated testing practice in the organization. The end to end call establishment process is one of the most critical and complex processes, which could be automated first with ideal execution process and exceptions could be added progressively.

There are several test automation tools available and one could be selected according to the requirement of the PharmaServe. Some of the test automation tools are Selenium, Katalon Studio, Unified Functional Testing, Test Complete, SoapUI.

Although automated testing brings several benefits for the organization, the usability testing could be manual or some usability modeling things could be engaged.

Recommendation - 6: Introduce Continuous practices

While continuous practices can have many forms such as continuous integration, continuous delivery and continuous deployment, one of the goals of continuous practices for an organization is to have more control on the deployment pipeline so that the product is either deployed on production automatically or the organization has the capability to deploy at any time according to their business need. The delivery pipeline could be fully automated so that the developed product is deployed directly to the production through some staging processes or involve some manual process which but providing the capability so that the deployment could be done at any time according to the needs of the organization.

There are several benefits of implementing continuous practices in an organization such as PharmaServe, which is engaged in delivering services through some software platform. From a literature study [16][17][18][19] the following benefits of implementing continuous practices are summarized.

- Enable organizations to release its product or service features more frequently, because the build time and test time are significantly reduced. An organization can reduce its go to market time which can give it a competitive advantage in the operating market.
- Continuous practices increase the visibility on the deployment process, which enables developers to receive faster and frequent feedback on the issues in the delivery pipeline, as a consequence the earlier detection of the faults are possible. It has a significant impact on the product quality and customer satisfaction.
- Continuous practices enhances the dependability and reliability of the deployment process.

6 TO-BE DELIVERY PIPELINE

In this section I describe the to-be delivery pipeline for the organization. The to-be delivery pipeline with possibility of automation using certain tools, techniques and processes are explained here. The outcome of relevant literature studies are presented here. Finally, an analysis is shown how the to-be delivery pipeline could be evaluated to validate if it brings the expected outcome.

While tools and components to implement an automated delivery pipeline are directly related with the many factors including the product itself, studies [20] identified certain categories of tools and approaches, such as automatic commit and deploy with Jenkins, automated testing with selenium, version control with Git, containerization with Docker, and configuration management with Puppet and Chef. Containerization is required when the development and deployment environment varies. And configuration management with Puppet and Chef is required to be considered when the system requires to be deployed on several systems. Although some subsystems of the videocon system such as mobile app, kiosk app will be deployed on many systems but these are mainly controlled from structured release system such as App Store.

In addition to the appropriate tools and components, to have an efficient CI/CD system in an organization the application must have considered certain Architecturally Significant Requirements (ASR), while developing the system. These are deployability, security, loggability, modifiability, monitorability and testability [21]. Four out of these six ASRs are considered to some extent to meet the functional and business requirements of the organization, but there are still opportunities in the VideoCon system to enhance its loggability and monitorability ASRs. Loggability is the capability of an application to log the critical actions in the system. Monitorability means the capability to monitor certain actions while the system is in production. These could be considered in the way of product maturity process, however these are not considered as major obstacles to start with.

Considering the recommended enhancements in chapter 4, the to-be delivery pipeline for the PharmaServe is shown in picture.

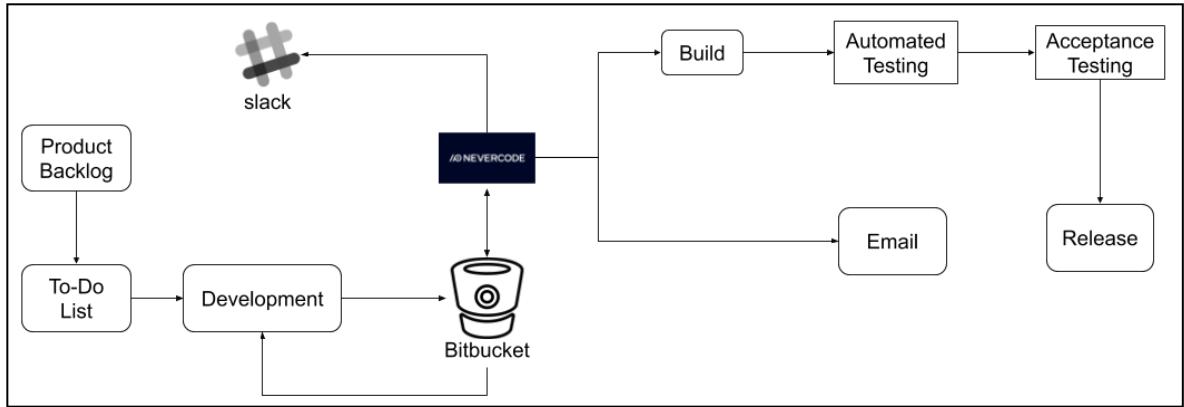


Figure 5: Recommended delivery pipeline

There are six different components which make this delivery pipeline different from the current one. In the subsequent section, how these can be implemented and what impact it could have will be discussed.

Bitbucket Repository: a source code repository is essential for any organization who has any kind of connection with any software product. Bitbucket offers unlimited private repositories. Repositories can be created for each of the products and access could be given to the developers. Creating a bitbucket repository is quite simple. The challenges the organization could face is to introduce the practice of checking in the source code to the repository. It might create an additional task for the developers if they already have to manage their repositories.

Testing Lab: The testing lab would enable the organization to see the product in its current development status. Currently it involves inefficient efforts to change the identification of the devices so that they could be used as test devices, then download the builds and finally test the products. However if the testing lab is there, the devices could be marked with some identification and with the help of automatic build process the build would be available to these marked devices. Going forward, the organization could enhance the environment to demonstrate the product as it is in the production, which would help to

simulate an error from the complaints received from customers or users. PharmaServe need to invest on some initial infrastructure development and setting up the system. The challenges of having a testing lab is to have an exhaustive collection of the devices in the market.

Automated Testing: Automated testing is one of the prerequisites for continuous practices. There are several tools such as Appium, available for introducing automated testing for Ionic, Cordova and Firebase applications. The major challenge PharmaServe will face to introduce unit testing and integration testing in all of its already developed source code. The price of additional development could be considered as an investment for the quality and process enhancement for the product.

CI/CD System: One of the core things for a delivery pipeline to be automated in an organization is to have a continuous integration server such as Jenkins. Jenkins is one of the popular, open source, and free CI server. For a fully functional CI/CD system the product architecture should consider some concerns. The delivery pipeline for a traditional web application which is deployed on a web server could be configured in Jenkins with less effort. Mobile applications such as android or iphone have their own release build and management processes. The developed app is uploaded to App Store or Play Store, from there the apps are released for alpha testing, internal testing, beta testing and finally releasing to the actual users. Integration of such system with Jenkins are quite cumbersome as it requires using several plug-ins and has a possibility of making the entire process complicated to manage. Considering this issue, after studying different options, it appears that a cloud based CI/CD system such as nevercode could be a suitable option for a startup organization such as PharmaServe. Nevercode provide easy integration of Ionic, Cordova, Firebase based applications and releasing to Play Store and App Store. The biggest challenge the organization could face is to management buy-in for such a system. A pilot test could be executed to see if the desired goals are achieved and then could be adjusted according to the requirements.

Automated Build: The build scripts are written and configured in the CI/CD system, so that whenever there is a code checked into the repository the build will be created automatically and available to the testing devices for testing.

Automated Release: The release processes are configured in the CI/CD system so that the delivery pipeline has a capability of releasing the product whenever the organization decides to do so.

In [22], continuous delivery practices have been implemented in a large financial organization and extracted the general practices, practices for managing the technical debts, testing practices in the continuous delivery pipeline. It was found that preferred methodology was Scrum by most of the respondents and at least the developers had clear product vision. To manage the technical debts some practices such as source code comments, code reviews, analysis of bad codes, and metrics were followed. Test Driven Development approach was followed in general.

[23] reported a proof concept for CI/CD for an insurance company. The technology stack used in the development and the organization of the activities are different from the PharmaServe. While the case organization involved itself into development, the development in the PharmaServe are conducted by the partner company, hence the tools used in the pipeline are not suitable for PharmaServe. However similar Proof of Concepts could be conducted for the PharmaServe.

7 DISCUSSION

A systematic literature study conducted by the author of this report shows that there is no study conducted for implementing continuous practices in an organization which is considered as a startup and outsources its software development activities to some other partner organization. Most studies were conducted for medium to large organizations and those which are directly involved in software development. Several challenges have been identified by different studies in implementing continuous practices and those challenges are analyzed here to see their validities in the PharmaServe.

[15] has implemented continuous practices for one book making organization and the challenges identified were categorized into three different aspects such as organizational, process related and technical. Organizational challenges are mainly due to the structure of the organization, which affects the decision making process such as releasing a product into production. Different units in the organization are involved with their different goals, which could limit the ultimate goal of implementing continuous practice. Process related challenges are due to the nature of a process execution, for instance change management process involves a change control board, hence each time a change is introduced to the product it creates some management overhead which in turn impacts the delivery pipeline. Study showed that, tools and techniques for implementing such practices are not widely available and selecting some tool can create vendor lock-in.

[16] conducted a systematic literature review on continuous practices and have identified the following challenges as reported by different other studies. The challenges are summarized in the table.

Common Challenges
<ul style="list-style-type: none">● Team awareness and communication, such as<ul style="list-style-type: none">○ lack of awareness and transparency among the team members

<ul style="list-style-type: none"> ○ coordination and collaboration ● Lack of investment <ul style="list-style-type: none"> ○ cost of introducing new tools ○ scarcity of expertise and required skills ○ indirect pressure and workload for team members ○ lack of appropriate tools ● Change resistance <ul style="list-style-type: none"> ○ general resistance to change ○ scepticism and distrust ○ Organizational structure, processes and policies 		
CI Challenges	CDE Challenges	CD Challenges
<ul style="list-style-type: none"> ● Testing <ul style="list-style-type: none"> ○ Lack of test strategy ○ poor test quality ● Merging conflicts 	<ul style="list-style-type: none"> ● Lack of suitable architecture <ul style="list-style-type: none"> ○ dependencies in design and code ○ database schema changes ● Team dependencies 	<ul style="list-style-type: none"> ● Customer challenge <ul style="list-style-type: none"> ○ customer environment ○ dependency on legacy systems ○ customer preference ● Domain constraints

Table - 2: *Summarized challenges of continuous practices*

Challenges during the change

Adapting agility such as continuous practices in an organization is fully related to the prevailing culture of that organization and such cultural changes are time consuming and expensive process. But adapting such changes at an earlier phase of the organizational growth will incur less cost of changes than that of later.

Limitation of this study

The biggest limitation of this study is that, it has been conducted based on literature reviews from different research databases. Most of the studies found in the current literature do not reflect the case of a startup organization, who is dependent on a development partner for the most critical product. However the recommendations could be utilized to conduct an experiment and the outcome could be analyzed to have more reliable recommendations and generic to similar kind of organizations.

8 CONCLUSIONS

This study was attempted to understand the environment of a startup business, which is involved in delivering digital services but due to the limitation of resources need to rely on a development partner for its core product to be developed. The target was to come up with some recommendations which could be utilized towards the achievement of an automated delivery pipeline.

The current delivery pipeline including all relevant components within the organization has been captured. Relevant literature on delivery pipeline, environment of startup organization have been reviewed to connect this study with current research trends. I have come up with the recommendations with proper reasoning behind those, explained how these could be implemented, and how they could enhance the business goals for the organization. The challenges the organization might face while implementing these were also discussed.

REFERENCES

1. Pofeldt, E. (2013, June 04). U.S. Entrepreneurship Hits Record High. Retrieved from
<https://www.forbes.com/sites/elainepofeldt/2013/05/27/u-s-entrepreneurship-hits-record-high/#37c61651d797>
2. Leybourne, S. A. (2009). Improvisation and agile project management: A comparative consideration. *International Journal of Managing Projects in Business*, 2(4), 519-535. doi:10.1108/17538370910991124
3. Jha, M. M., Vilardell, R. M., & Narayan, J. (2016). Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market. *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*. doi:10.1109/icgse.2016.24
4. Giardino C, Unterkalmsteiner M, Paternoster N, Gorschek T, Abrahamsson P. What do we know about software development in startups? *S-M.* 2014;31(5):28-32. <https://ieeexplore.ieee.org/document/6898758>. doi: 10.1109/MS.2014.129.
5. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45-77. doi:10.2753/mis0742-1222240302
6. Tyrväinen, P., Jansen, S., & Cusumano, M. A. (2010). *Software business: First International Conference, ICSOB 2010, Jyväskylä, Finland, June 21-23, 2010: Proceedings*. Berlin: Springer-Verlag.
7. Valtakoski, A., & Rönkkö, M. (2010). Diversity of Business Models in Software Industry. *Lecture Notes in Business Information Processing Software Business*, 1-12. doi:10.1007/978-3-642-13633-7_1
8. Blank, S. G., & Dorf, B. (2012). *The startup owners manual: The step-by-step guide for building a great company*. Pescadero, CA: K & S Ranch.

9. Ries, E. (2017). *The lean startup: How todays entrepreneurs use continuous innovation to create radically successful businesses*. New York: Currency.
10. Crowne, M. (n.d.). Why software product startups fail and what to do about it. Evolution of software product development in startup companies. *IEEE International Engineering Management Conference*. doi:10.1109/iemc.2002.1038454
11. Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., & Abrahamsson, P. (2014). What Do We Know about Software Development in Startups? *IEEE Software*, 31(5), 28-32. doi:10.1109/ms.2014.129
12. Gartner (October 08, 2014). Gartner Identifies the Top 10 Strategic Technology Trends for 2015 [Online]. Available: <http://www.gartner.com/newsroom/id/2867917>
13. Rahman, A. A., Helms, E., Williams, L., & Parnin, C. (2015). Synthesizing Continuous Deployment Practices Used in Software Development. 2015 Agile Conference. doi:10.1109/agile.2015.12
14. Hanna R., Raffo D., Daim T.U. (2014) Managing Issues of IT Service Offshore Outsourcing Projects. In: Daim T., Neshati R., Watt R., Eastham J. (eds) *Technology Development. Innovation, Technology, and Knowledge Management*. Springer, Cham
15. Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2019). *Fundamentals of Business Process Management*. Berlin: Springer Berlin.
16. Chen, L. (2015). Continuous Delivery: Huge Benefits, but Challenges Too. *IEEE Software*, 32(2), 50-54. doi:10.1109/ms.2015.27
17. Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5, 3909-3943. doi:10.1109/access.2017.2685629
18. Leppanen, M., Makinen, S., Pagels, M., Eloranta, V., Itkonen, J., Mantyla, M. V., & Mannisto, T. (2015). The highways and country roads to continuous deployment. *IEEE Software*, 32(2), 64-72. doi:10.1109/ms.2015.50

19. Leppanen, M., Kilamo, T., & Mikkonen, T. (2015). Towards Post-Agile Development Practices through Productized Development Infrastructure. *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*. doi:10.1109/rcose.2015.14
20. Agarwal, A., Gupta, S., & Choudhury, T. (2018). Continuous and Integrated Software Development using DevOps. *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. doi:10.1109/icacce.2018.8458052
21. Chen, L. (2015). Towards Architecting for Continuous Delivery. *2015 12th Working IEEE/IFIP Conference on Software Architecture*. doi:10.1109/wicsa.2015.23
22. Vassallo, C., Zampetti, F., Romano, D., Beller, M., Panichella, A., Penta, M. D., & Zaidman, A. (2016). Continuous Delivery Practices in a Large Financial Organization. *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. doi:10.1109/icsme.2016.72
23. Soni, M. (2015). End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. *2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. doi:10.1109/ccem.2015.29
24. Burke, G. J., Carrillo, J. E., & Vakharia, A. J. (2007). Single versus multiple supplier sourcing strategies. *European Journal of Operational Research*, 182(1), 95-112. doi:10.1016/j.ejor.2006.07.007
25. Newman, R. G. (1989). Single Sourcing: Short-Term Savings Versus Long-Term Problems. *Journal of Purchasing and Materials Management*, 25(2), 20-25. doi:10.1111/j.1745-493x.1989.tb00478.x
26. Treleven, M., & Schweikhart, S. B. (1988). A risk/benefit analysis of sourcing strategies: Single vs. multiple sourcing. *Journal of Operations Management*, 7(3-4), 93-114. doi:10.1016/0272-6963(81)90007-3
27. Veretennikova, N., & Vaskiv, R. (2018). Application of the Lean Startup Methodology in Project Management at Launching New Innovative Products.

- 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT).*
 doi:10.1109/stc-csit.2018.8526731
28. Duvall, P. M., Matyas, S., & Glover, A. (2013). *Continuous integration improving software quality and reducing risk*. Upper Saddle River, NJ: Addison-Wesley.
 29. Meyer, M. (2014). Continuous Integration and Its Tools. *IEEE Software*, 31(3), 14-16. doi:10.1109/ms.2014.58
 30. Weber, I., Nepal, S., & Zhu, L. (2016). Developing Dependable and Secure Cloud Applications. *IEEE Internet Computing*, 20(3), 74-79. doi:10.1109/mic.2016.67
 31. Continuous Delivery. (n.d.). Retrieved from <https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/>
 32. 2015 State of DevOps Report. (n.d.). Retrieved from <https://puppet.com/resources/whitepaper/2015-state-devops-report>
 33. Fitzgerald, B., & Stol, K. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176-189. doi:10.1016/j.jss.2015.06.063
 34. What is Continuous Deployment & Continuous Delivery? (n.d.). Retrieved from <https://electric-cloud.com/resources/continuous-delivery-101/continuous-deployment/>
 35. Nie, E., & Hammouda, I. (2017). An Exploratory Study on Strategic Software Development Outsourcing. *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. doi:10.1109/icgse.2017.10
 36. Kramer, T., Klimpke, L., & Heinzl, A. (2013). Outsourcing Decisions of Small and Medium-Sized Enterprises: A Multiple-Case Study Approach in the German Software Industry. *2013 46th Hawaii International Conference on System Sciences*. doi:10.1109/hicss.2013.441

37. Faiz, M. F., Qadri, U., & Ayyubi, S. R. (2007). Offshore Software Development Models. *2007 International Conference on Information and Emerging Technologies*. doi:10.1109/iciet.2007.4381329
38. Dubé, L., & Robey, D. (1999). Software stories: Three cultural perspectives on the organizational practices of software development. *Accounting, Management and Information Technologies*, 9(4), 223-259.
doi:10.1016/s0959-8022(99)00010-7
39. Sommerville, I. (2016). *Software engineering*. Boston: Pearson.
40. Bourque, P. (2014). *Guide to the software engineering body of knowledge Version 3.0 ; SWEBOK ; a project of the IEEE Computer Society*. Los Alamitos, Calif.: IEEE Computer Soc.
41. Atlassian. (n.d.). Jira | Issue & Project Tracking Software. Retrieved from <https://www.atlassian.com/software/jira>
42. Atlassian. (n.d.). Confluence - Team Collaboration Software. Retrieved from <https://www.atlassian.com/software/confluence>
43. Atlassian. (n.d.). The Git solution for professional teams. Retrieved from <https://bitbucket.org/>
44. Bitbucket Support. (n.d.). Retrieved from <https://confluence.atlassian.com/bitbucketserver0414/using-branches-in-bitbucket-server-895367626.html>
45. Ionicframework. (n.d.). Ionic - Cross-Platform Mobile App Development. Retrieved from <https://ionicframework.com/>
46. (n.d.). Retrieved from <https://firebase.google.com/>
47. Get Started Fast. (n.d.). Retrieved from <https://cordova.apache.org/>
48. Communication APIs for SMS, Voice, Video and Authentication. (n.d.). Retrieved from <https://www.twilio.com/>
49. Cloud Functions - Event-driven Serverless Computing | Cloud Functions | Google Cloud. (n.d.). Retrieved from <https://cloud.google.com/functions/>
50. Kersten, M. (2018). What Flows through a Software Value Stream? *IEEE Software*, 35(4), 8-11. doi:10.1109/ms.2018.2801538

51. Gmeiner, J., Ramler, R., & Haslinger, J. (2015). Automated testing in the continuous delivery pipeline: A case study of an online company. *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. doi:10.1109/icstw.2015.7107423