SELECTION OF CONTINUOUS INTEGRATION TOOLS FOR AGILE METHODOLOGY: AN ANALYTIC HIERARCHY PROCESS (AHP) APPROACH

FATMA KEL

JANUARY 2021

SELECTION OF CONTINUOUS INTEGRATION TOOLS FOR AGILE METHODOLOGY: AN ANALYTIC HIERARCHY PROCESS (AHP) APPROACH

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED

SCIENCES OF

ÇANKAYA UNIVERSITY

BY

FATMA KEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

DEPARTMENT

JANUARY 2021

ABSTRACT

SELECTION OF CONTINUOUS INTEGRATION TOOLS FOR AGILE METHODOLOGY: AN ANALYTIC HIERARCHY PROCESS (AHP) APPROACH

KEL, Fatma

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Gül TOKDEMİR

January 2021, 54 pages

In the software world, it is known that continuous integration technology, importance of which is growing rapidly, enhances the efficiency of projects and saves significant time to the user. Institutions that adopt agile software methodology want to include the continuous integration, which is part of agile software, into their business lives. Continuous integration provides automation of various workloads such as version control of the projects, build issues, problems after build, running and reporting of tests. In this way, time to be spent on these tasks and unnecessary workload loss is reduced. Thus, by ensuring more effective use of employees, the efficiency of projects is significantly improved. This technology, which is very popular, brings along a lot of competition in the market. There are a number of continuous integration tools on the market and all of them have different characteristics they offer to get ahead of each other. Choosing the right tool among the alternatives is a challenging task for the companies. Especially for an institution with a closed network such as TÜBİTAK SAGE, it is difficult to choose and make the right choice for a continuous integration tool. The closed network brings the obligation to keep the repositories of the tools to be used in the local network. Therefore, if a tool does not meet the expected requirements and cannot be used, it will cause material and moral damage rather than a benefit. If the wrong tool is selected, then the repositories created for a new tool need to be reinstalled from the beginning. When the continuous integration tool is chosen wrongly, switching to a new technology will cause loss of time and demotivation of the employees. Hence, the expectation of increase in productivity by the usage of such tools will diminish and will bring inefficiency instead. There is no study in which the AHP method is used for continuous integration tools in literature. The difference of the AHP process applied in this study is also a part of the originality of the study. In line with all these, in this study, it is aimed that a continuous integration tool that is suitable for TÜBİTAK SAGE culture and working style is selected. This selection is performed using Analytic Hierarchy Process, which is a multi-criteria decision making method. A systematic process was applied with the consortium of the study group and the expert group. As a result, criteria set consisting of compatibility, flexibility and expandability, functionality and reliability were determined to be used when selecting continuous integration tools. As a result, the priority levels of these criteria were functionality with the highest value with 0.33427, while compatibility with the lowest value with 0.19172. Finally, four different continuous integration tools selected according to the corporate culture and requirements were evaluated with the AHP method. As a result, among the alternative, alternative 1(Jenkins) was chosen first with rate of 37,14%, followed by Alternative 2(TeamCity) with rate of 28,86%.

Keywords: Continuous Integration, Agile Methodology, Continuous Integration tools, Multi-Criteria Decision Making, AHP.

ÖZ

ÇEVİK METODOLOJİDE SÜREKLİ ENTEGRASYON ARAÇLARININ SEÇİMİ: ANALİTİK HİYERARŞİ SÜRECİ (AHP) YAKLAŞIMI

KEL, Fatma

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Dr. Öğr. Üyesi Gül TOKDEMİR

Ocak 2021, 54 sayfa

Yazılım dünyasında önemi hızla artan sürekli entegrasyon teknolojisinin, projelerin verimliliğini arttırdığı ve kullanıcıya önemli bir zaman kazandırdığı bilinmektedir. Çevik yazılım methodolojisini benimseyen kurumlar, çevik yazılımın bir parçası olan sürekli entegrasyonu da bir şekilde hayatlarına katmak istemektedirler. Sürekli entegrasyon, projelerin versiyon kontrolleri, derlenmeleri, derlenme sonrası çıkan sorunlar, teslerin koşulması ve raporlanması gibi bir çok iş yükünün otomatikleştirilmesini sağlamaktadır. Bu sayede, bu işlere hacanacak zamandan ve gereksiz iş yükü kaybından tasarruf edilmektedir. Ayrıca, çalışanların daha verimli bir şekilde kullanılması sağlanıp, projelerin verimliliği ciddi bir sekilde arttırılmış olmaktadır. Çok fazla rağbet gören bu teknoloji, piyasada çok fazla rekabeti beraberinde getirmektedir. Piyasada çok sayıda sürekli entegrasyon aracı ve hepsinin birbirlerini geçebilmek için öne sundukları farklı becerileri mevcuttur . Bu kadar araç arasından da seçim yapmak oldukça zorlaşmaktadır. Özellikle TÜBİTAK SAGE gibi kapalı ağa sahip bir kurum için seçim yapmak zordur ve dogru seçimi yapmak oldukça önem arz etmektedir. Kapalı ağ, beraberinde kullanılacak araçların deposunu yerel ağda tutma zorunluluğunu getirmektedir. Dolayısıyla bir aracın beklenen isterleri karşılamaması ve kullanılamaması, faydadan çok maddi ve manevi zarara neden olacaktır. Yanlış bir araç seçilirse, yeni bir araç için oluşturulan depo sil baştan tekrar kurulacaktır. Bütün bunlar göz önünde bulundurulduğunda, yeni bir teknolojiye alışma

evresinde vakit kaybı ve çalışcan motivasyon kayıpları ortaya çıkacaktır. Bununla birlikte artması gereken verimlilik, tam tersi yönde etkisini gösterecektir. Litaretürde Sürekli entegrasyon araçları için AHP methodunun kullanıldığı bir çalışmaya rastlanılmamıştır. Bu çalışmada uygulanan AHP sürecinin farklılığı da çalışmanın özgünlüğünün bir parçasıdır. Bütün bunlar doğrultusunda, bu çalışmada, TÜBİTAK SAGE kültürüne ve çalışma stiline uygun sürekli entegrasyon aracı seçilmesi sağlanmaktadır. Bu seçim çok kriterli karar verme yöntemi olan Analitik Hiyerarşi Süreç metodu kullanılarak yapılmaktadır. Bu çalışmada, çalışma grubu ve uzman grubun konsorsiyumu ile sistematik bir süreç uygulanmıştır. Sonuç olarak, sürekli entegrasyon araçları seçilirken uyumluluk, esneklik ve genişletilebilirlik, işlevsellik ve güvenilirlikten oluşan kriter seti kullanılacağı belirlenmiştir. Çalışmada, en yüksek değere sahip kriter 0.33427 öncelik derecesi ile işlevsellik olarak belirlenmiştir. En düşük değere sahip kriter ise 0.19172 öncelik derecesi ile uyumluluk olmuştur. Son olarak, kurum kültürümüze ve isterlerine uygun dört farklı sürekli entegrasyon aracının Analitik Hiyerarşi Süreç sistemi ile değerlendirilmesi gerçekleştirilmiştir. Çalışmanın sonucunda, alternatifler arasından Alternatif 1 %37.14 oranı ile birinci seçilirken, onu %28.86 ile Altenatif 2 takip etmektedir.
Anahtar Kelimeler: Sürekli Entegrasyon, Çevik Metodoloji, Sürekli Entegrasyon Araçları, Çok ölçütlü karar verme, AHP.

ACKNOWLEDGEMENTS

First and foremost, I am extremely grateful to my supervisor, Assist. Prof. Dr. Gül TOKDEMİR for their invaluable advice, continuous support, and patience during my thesis. Her immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

I wish to thank the examining committee for their kindness during the presentation of this thesis.

I would like to thank TÜBİTAK SAGE, which offers the opportunity to benefit from all kinds of opportunities by supporting my thesis project.

Finally, I would like to express my deep gratitude to my father Ömer KEL, my mother Lütfite KEL, and my sister Gökçe KOÇAK, for their endless and continuous encourage and support throughout these years. They always stood beside me all the way. All acquirements, which I have gained until now, have occurred under favour of their limitless and endless support, reliance and guidance. Thank you once again to my family for making feels me that I have a great family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CI	Continuous Integration
AHP	Analytic Hierarchy Process
ANP	Analytic Network Process
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
EEE	Electrical Electronics Engineering
ME	Mechanical Engineering
CENG	Computer Engineering

CHAPTER 1

INTRODUCTION

In today's world where everything is racing with time, time and speed factors have an important place in the software industry. Henry Ford emphasized the importance of time with the following words: "It has been my observation that most people get ahead during the time that others waste" [1]. For this reason, companies in the software sector compete with time and want the products to come out quickly. The impatience of customers and the prolongation of the schedule of projects put serious pressure on software developers and managers. One of the external qualities that stakeholders expect from software developers is speed [2][3].

Agile methods have started to be adopted in the name of speed and dynamism in Software Engineering. One of the 12 principles of agile software methodology is "Our Highest Priority is to satisfy the Customer through Early and Con-tinuous Delivery of Valuable Software" [4]. Agile software development methodology requires fastness. This methodology determines the points to be considered while being fast and shortens your project time. In this way, it enables the tests to be carried out while providing faster and easier communication [5][6]. In institutions that have adopted the agile software methodology, there are cases of teamwork, weekly maybe daily code additions to the project, and the control of the added codes due to the continuous development. It is an important that these situations do not affect the working system.

In projects that are developed by software developer teams, many team members make changes on the code, write separate modules which are integrated to the project. These changes and additions affect the whole system. The smooth running of the system despite all the additions and changes is an important factor. Even the slightest mistake that occurs as a result of changes and additions causes the project process to be prolonged. Project team members spend most of their time looking for the module that the error originated from rather than developing the project and making the project work. Detecting the problematic parts of the code is of great importance in terms of speed in large project. With all these in mind, automating the software process of the project will ensure that the problem is solved in an efficient, safe and functional manner [7].

Software companies are in a fierce race to improve and make their development processes more efficient. Companies have to constantly adapt to new developments to gain a competitive advantage. This need for rapid development and adaptation has led to the DevOps culture and philosophy, which is a more holistic approach with continuous integration (CI) and continuous delivery (CD) [8]. DevOps emerged to eliminate the bottleneck preventing development teams from delivering to operations faster and more frequently. DevOps approach requires a strong and continuous connection between development and operations teams [9]. Continuous integration is at the heart of the entire DevOps lifecycle.

Continuous integration is defined as “software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible” [10].

Continuous integration is one of today's technologies that ensure both quality work and minimization of time loss in large software projects [11]. In this way, the risk of generating errors in large projects, where team members develop various software modules, can be minimized. When an error occurs, the module in which the error occurs is communicated to the authorized person, eliminating the search for the origin of the error and providing the developer more time and speed gain, and thus more focus on writing quality code [12].

Continuous integration is the name given to the automatic processing of the written code and the developed module while being included in the project. Continuous integration is a system that checks that the entire system is operational after every change made to the code. It is also the method used to determine if the change caused breaks in some parts of the system. All actions such as downloading dependencies from repos, including them in the project, passing the developed code snippet through tests (unit, integration, etc. ...), informing the developer or the responsible person according to possible situations (push notification, sms, mail, etc.), packaging the work, usually takes place in our lives as part of this concept [13][14].

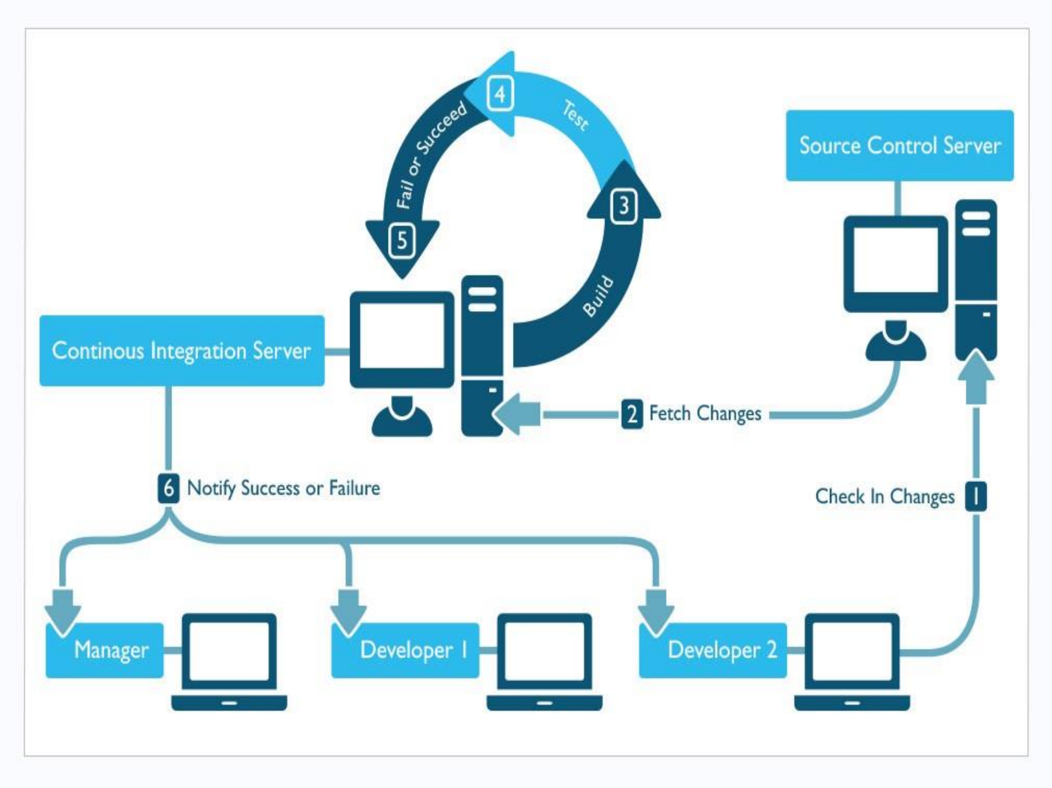


Figure 1 CI and DEVOPs

Continuous integration is the method used to determine if the entire system is operational after each change in the code, and that the change does not cause breaks in some parts of the system. Unit tests are needed to detect breakages [11]. These tests are run automatically after a new build is prepared as a result of the change. Since the change is a part of the new structure, errors in the tests mean that the change made broke the system. All programmers are informed about this situation and it is ensured that the error is eliminated as soon as possible and the tests always give positive results. With continuous integration, as a result of the work done on the code by the programmers, working version is created always.

In this context, there are many CI tools with diverse features. In order to avoid problems in the future project integration and process, institutions should determine the criteria that they expect from continuous integration. They should try to choose the right tool that meets these criteria in the most optimal way. In institutions, the selection of new tools is usually made by the manager, that is, with the initiative of only one person or by the supervisors who will use the tool very little. However, this choice is a complex problem with multiple criteria. In this study, AHP method, which is one of the multi-scale decision making methods, is used.

Many criteria should be taken into consideration in the selection of CI devices. We can list these criteria as follows; functionality, compatibility, reliability, long life availability, flexibility, extensibility, be open source or commercial. The abundance of criteria and alternatives led us to multi-scale selection methods [15]. In this study, it was decided to use Analytical Hierarchy Process, one of the multiscale selection methods. AHP is one of the Multi Criteria Decision Making methods put forward by Thomas L. Saaty that helps the decision maker [16]. AHP has been used in many areas and in the software field in the decision-making process in multi-criteria situations, and it has been effective in making an optimal decision for the user.

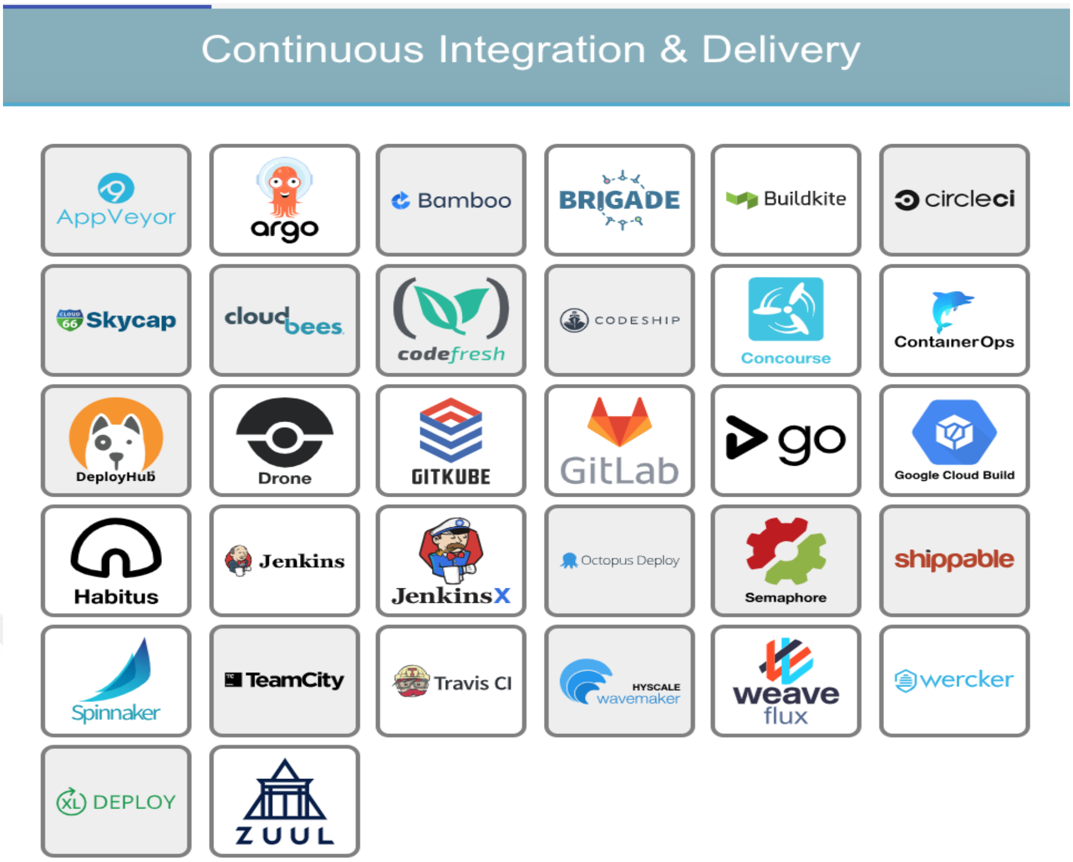


Figure 2 CI TOOLs

The features and criteria provided by CI devices will be explained in detail in section 2.

1.1. The Multi-Criteria Decision-Making

Multi-criteria decision making (MCDM) is defined as the selection process of the decision maker using at least two criteria within a set of countably finite or uncountable choices [17]. Since there are many CI tools on the market and there are too many features and criteria, it is difficult to find a tool that meet the desired criteria. In this case, Multi-criteria decision making (MCDM) methods could be employed for determining the criteria and for deciding the tool.

The use of continuous integration has become widespread around the world. Automation of the software development process to quickly release software implementation is an important growth factor for the continuous integration tools market. As the time to market accelerates, organizations aim for software updates to be released quickly to the market. That's why continuous integration tools help organizations increase developer productivity. There are more than 50 continuous integration tools in line with this demand in the market [18]. According to a report, it is stated that the growth in continuous integration will continue rapidly in the coming years [19]. Figure 3, shows the estimated increase of the continuous integration tools sector over the years according to the continents.

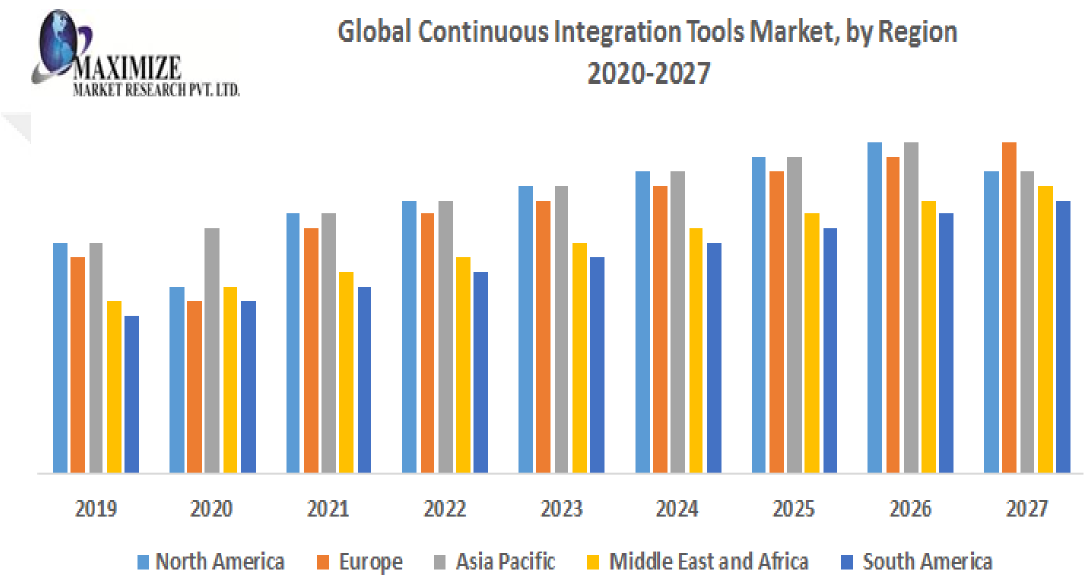


Figure 3 Global Continuous Integration Tools Market by Region

The competition of the continuous integration tools and the features they provide to the user vary. Institutions should choose tools according to their own expectations. However, it is very difficult to make the right choice among so many options. It is possible to make a choice where the expectations or demands are not fully met. In this case, it is an expected result to not be able to fully use the features offered by the continuous integration tool and not to benefit from the tool sufficiently. As a result, there will be no gain in software product quality and time. On the contrary, there will be a loss of motivation and time by team members during the adaptation period.

Laukkanen et al talks about the difficulties encountered in continuous integration in his study [20]. These challenges are presented as build design, system design, integration, testing, and release, human and organizational, resource. So, these criteria should be taken into consideration in CI tools selection.

In this study, multi-criteria decision making method will be used as a decision-making method due to the existence of criteria and many tools.

CHAPTER 2

BACKGROUND AND RELATED WORKS

2.1. The Importance of Continuous Integration in Software Processes

While agile methods can reach a more frequent software development pace, a bottleneck has emerged in organizations because the Operations function (Ops) coordinating the software is often not compatible with the Development function(Dev) [21]. Bottleneck causes long delays in software releases available to customers. To solve this problem, Debois advocated a better integration between Dev and Ops functions called DevOps [22][23].

Continuous integration is a software development application where team members regularly integrate their work, automate testing and verification [11]. In this way, CI applications help development team's productivity by freeing the team from manual tasks, removing complexities and focusing on re-introduction of new features in software. The main benefits of CI implementations are to reduce risk of error and make the software error-free and reliable, eliminating barriers to frequent delivery [24]. Through CI applications, teams who are aware of the effects of changes in the code can quickly intervene and refresh the code [13]. Implementing CI increases the speed of the entire team, including feature release and bug fixes, hence tasks that lasted weeks and months can now be done in days or even hours. Makam emphasized that when CI is applied, code quality improves and updates can be done quickly [25].

In addition, the features provided by continuous integration applications can be listed as faster builds, integration, quick and easy setup, container support, testing with multiple runtimes, versions, and environments, integrated code coverage and test results visualization, and flexible infrastructure options [26]. Providing all these features together increases the speed and efficiency of the project [12][27]. Arachchi et al. have demonstrated the utility of CI in projects in their study [24]. Brandtner et al. have mentioned that the integration of a modern CI application into the development process of a software project is fully automated and its execution is triggered after each action [28].Developers or testers usually detect the CI process only in the event of a compile interruption or test failure. If this happens, they will be notified by an automatically generated notification, for example by e-mail. Brandtner et al. highlight that CI will help detect and fix problems as early as possible during such exception-driven behavior, integration runs [28].

Parameters such as improving product quality, improved customer satisfaction, reliable release, improved productivity, functionality and efficiency are the main advantages that motivate organizations to invest in CI [24].

2.2. Importance of choosing the right CI tools

Nowadays, software companies understand the value of continuous integration, and because continuous integration increases the software speed, hence the quality and reduces the cost, companies have turned to this area. The high demand has led to the proliferation of continuous integration tools and increased competition. The fact that there are many tools and the competition between them made the tools differ in many criteria such as their features, price, functionality, flexibility, expandability, compatibility, usability, safety and longevity. Some of the tools are open source and some are paid. Just because the product is expensive does not mean it is better than a low cost or open source tool. Generally, all of these tools try to automate the software creation process. Despite this fact, continuous integration tools have their own strengths and weaknesses, so choosing the right tool for a job is really important.

Wrong choice can reduce overall flexibility; it can increase the time required to perform ordinary actions or cause some problems in the continuous integration process. A decision that is not made right from the beginning will increase the whole adjustment process, employees' adaptation to the tool and the integration process of projects.

In line with the increasing demand, there has been competition in the market in this area and a lot of continuous integration tools have emerged. Each strives to stay in the market with extra features to compete with others. At this stage, it is of great importance for the institutions to choose the right tools [15].

In this direction, features that should be evaluated in continuous integration tools are faster builds, integration, quick and easy setup, container support , testing with multiple runtimes, versions, and environments, integrated code coverage and test results visualization which can be listed as flexible infrastructure options [26]. In addition to these, it will be very difficult to choose a tool that meets many criteria, such as functionality, compatibility, reliability, long-term availability, flexibility, extensibility, availability, open source or commercial, and can accommodate them among all options [15][25][29]. On the other hand, a wrong decision will be slow and weary. It is important that the CI tool is compatible with the build automation tool used in the projects, an add-on to the test automation tool and is flexible and extensible for future features [24][25][27]. In the event of a malfunction in these features, the tool used to accelerate the software process will have the opposite effect, causing the process to slow down, delay, or make no progress, and will not serve its real purpose. If there is a problem with at least one of these, the tool selection process will start over and this will be weary. For the user, getting used to something new, breaking the order is a challenge in itself, and the problems that come on top of it can cause the breaking of the resistance. This problem will be even more cumbersome if a closed network is used as in the institution where this study will be conducted. The reason for this is that since the intranet is used, you have to download the plugins that the tool will use and set up your own repo, and this is not an easy process. On top of that, preparing everything from scratch in case of a problem, establishing a new repo will be financially and morally wearing. Although not all CI devices have open source code, it will be a wasted investment in the direction of wrong selection. Considering all these, it is thought that choosing the right tool will provide financial and moral benefits for the institution and the user in addition to its contribution to the projects. Polkhovskiy, compared CI tools in his study and underlined the importance of correct selection [15].

In this study, Analytical Hierarchy Process (AHP) which is one of the multiscale selection methods, is chosen for the selection of right CI tool for quality of software process, correct management of time and increasing efficiency in projects.

2.3. Analytical Hierarchy Process (AHP)

As mentioned in the previous section, it is important to make the right decision in choosing a new software system tool in case of an institution where closed network is used because of the difficult and time-consuming adaptation and installation process suitable for the closed network for each selected software tool. Therefore, the installation process of tools for the use of closed networks for every wrong decision software tool starts over. It is important in terms of time, efficiency and cost to complete the process. For this reason, it is decided to use analytic hierarchy process in the CI tool selection to be performed in this study.

The analytical hierarchy process (AHP) was first introduced by the Myers and Alpert duo in 1968, and in 1977, it was developed as a model by Saaty and made usable in the solution of decision making problems. AHP is a decision-making method used in the solution of complex problems involving multiple criteria. The most important feature of AHP is that it can include both objective and subjective thoughts of the decision maker in the decision process [30].

The analytical hierarchy process (AHP) is a powerful and flexible multi-criteria decision-making method introduced by Thomas L. Saaty that is applied to solve unstructured problems in a variety of decision-making situations, such as complex intensive decisions, including simple personal decisions [31]. One of the biggest advantages of this method is that it enables analytical evaluation of criteria without numerical values by means of comparison methods. AHP's application areas are quite wide. AHP is used for various situations such as establishment location applications, supplier selection, recruitment, selection of continuous improvement projects, hardware and software selection, Investment decision, determination of the most appropriate strategy, performance evaluation, and selection of product alternatives for consumers. The study of Thomas L. Saaty shows that it has been observed that the best decisions are made with AHP, which is used in many fields of health, military, education and many other sectors [32]. Yılmaz et al mentioned the use and importance of AHP in the field of health where vital decisions are taken [33]. In the field of software, Laplante emphasized the selection process of the software project management tool with AHP in her studies [31]. Li et al. proposed a framework for selecting software reliability criteria based on AHP theory and expert opinion, aiming to solve the problem of how to scientifically select appropriate software reliability criteria for software reliability engineering at each development stage [34]. In addition, in the software field, we can see that the study of Tekin uses the AHP method in the selection of measurement metrics in the software development process [35]. It can be seen in Ming-Chyuan Lin et al. study that AHP is also used in the customer-oriented product design process [36].

When we look at the studies in the literature above, we see that the AHP method is used in many areas such as education, military, health, software, etc. Although AHP has been used in software field recently, it is seen that it is not as common as other fields. In the light of these studies, the use of AHP has been deemed appropriate for tool selection in the continuous integration technology that has been decided to be used in a closed-network institution like TUBİTAK SAGE. Thus, it is aimed to be a guiding study in such tool selection in institutions. The aim of this study is to examine the selection of continuous integration tools with AHP method, which is one of the multi-criteria decision making methods. In this context, as an industrial case study in the TÜBİTAK SAGE software development group, weighting the selection criteria for continuous integration tool selection and evaluating the alternatives was carried out.

CHAPTER 3

METHODOLOGY

First of all, the problem of choosing the appropriate continuous integration tool was determined. After that, literature review related to MCDM and problem was searched. Then, alternatives and criteria were constructed according to the research results. Study group and expert group were determinated. Then, a questionnaire was applied to the study group. After criteria and alternatives were reduced by expert group according to survey results, AHP was applied by the contribution of expert group. The priorities of the criteria and alternatives were evaluated. Finally, the alternative with the highest priority was chosen as the final decision. Figure 4 shows all the steps of the case study process. Considering all of these, the research question of this study is: *What is the continuous integration tool that meets the criteria and features suitable for a closed networked institution?*

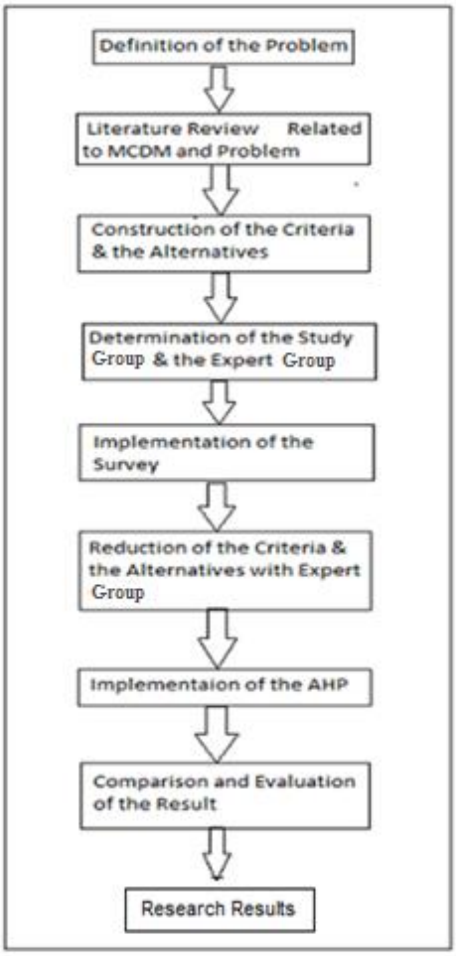


Figure 4 Step of the Case Study

3.1. Definition of the Problem

3.1.1. Determining User Expectations from Continuous Integration

A survey was conducted to software professionals' team who work in different software development tasks and fields under software coordination department and who is responsible from the transition to the continuous integration phase. The study group was asked to make multiple choices among the features of a continuous integration tool that are appropriate for their institution and working conditions. In line with the decisions of the team, the results obtained were evaluated by the expert group and 4 alternative tools that best suit the results were determined.

The features of the continuous integration tools were taken from the webpages of the tools and the table was created. The questionnaire for the study group was created in line with these features. The features that were researched and found on the pages were collected in the Table 1 [37]–[50].

Table 1 Continuous Integration Tools Features

Features Provided by Continuous Integration	**Description**
Built-In features	Built features supported language platform (C, C ++, java, C #, python..., and Providing build automation tool support (maven ant conan…)
Operating Systems	The operating systems with which the tool runs (Windows, Linux, MacOs, Unix...)
Integration and Software Support & Plugins	Comprehensive plugins & Integration support for platforms, UI, administration, source code management, build management (Ide, Jira, svn, github, parasoft…)
Container Support	Having a deployment plug-In or configuration for container editing tools like Kubernetes and Docker makes It easy for a CI tool to connect to the application’s target environment.
Docs and Supports	Documents and supporters provided to the user
Easy to use & setup	Ease of Installation and use
Use Case	Target project cases, small projects or large projects.
Support Continuous Delivery	Continuous Delivery Support Status(yes, no)
License Pricing	Free or price
Hosting Options	Hosting options Cloud or on premise or both
Open Source	The tool is open source or not

3.2. Identification of the study group and the expert group

3.2.1. Study Group

The study group consists of 41 software developers working in the software department. This group, which has a corporate culture, consist of people with a sense of responsibility and sufficient experience in the field of software.

3.2.2. Expert Group

The expert group members consist of software developers with titles such as team leader, unit manager, coordinator and / or chief scientist. Each of them has at least 5 years of working experience in software development. This group is responsible for any decision made in the field of software development process. Detailed information about the expert group is given in Table 5.

3.3. Implementation of the Survey

A meeting was held in the software development unit with the participation of the study group and the expert group. At this meeting, participants were informed about the continuous integration. Continuous integration, what it is, its advantages, disadvantages, purpose of use, has been mentioned comprehensively. In addition, it was announced that their demographic and work related information such as their work experience, areas of experience, graduation departments and work units will be collected. The survey is included in the appendix.

The survey has been prepared using Google forms. In the survey, the user was reminded of the general features of the continuous integration tools in a table. In addition, an explanation has been added to the options offered for selection. The analytical process format of the collected data was converted into graphic formats with the help of Google Sheets and Microsoft Excel. In this way, the results were evaluated in detail by presenting them to the expert group. According to the results of the survey, the criteria and alternatives were reduced by the expert group.

3.4. Determination of the Criteria & the Alternatives

According to the research conducted in this study, it was seen that there are more than 50 continuous integration tools in the market. These tools have been examined according to accessibility, sufficient technical features, usage rates and market evaluations. Alternatives must provide the necessary features in line with the needs of developers. It is seen that there are many criteria for continuous integration tools in the literature [15]. These criteria play an important role in the competition between continuous integration tools in the market which are listed in the Table 2 with their explanations. While evaluating continuous integration tools, the following criteria should be considered [15][25][29].

Table 2 Continuous Integration Tools Criteria

Criteria	Description
Functionality	CI reflects that the tool should meet all the requirements: Providing features such as Build execution, version Control System, security features built-in, User Interface.
FlexibiIity&Expandability	Great flexibility should be given too many different development methodologies also all the features a developer may need. The tool should be expandable and modifiable, allowing you to customize It to your needs.
Compatibility	Compatibility is a criterion that defines how well CI tool is integrated with other elements of development process.
Usability&Availability	It is the feature that the tool is easy to configure and use. Evaluation of the products offered in the market in terms of availability.
Reliability	Tool reliability feature in CI tool selection
Longevity	Longevity Is a factor that concerns the future of the tool.
Open Source&Commercial	There are many open-source and commercial software available on the market today.

The reduction process of the criteria has been made by the expert group, taking into account the results of the survey. In this way, the criteria set and alternatives set were finalized to apply AHP method.

3.5. AHP

In this study, AHP method was applied to an expert group of five participants. There are studies in the literature where AHP was applied in groups. Vaidya and Kumar stated that a study was conducted with an expert group of five participants in an AHP process for the exchange rate [51].Tekin et al. applied AHP to an expert group of six participants in their studies [52].

A decision-making problem solved with AHP has to go through some stages. These stages are explained below with their formulas.

Step 1: Definition of the Decision Making Problem

In the first step, the decision-making problem should be defined clearly. The suitability for analysis with AHP should be investigated. Defining the decision-making problem consists of two stages. In the first stage, decision points are determined. In the second stage, criteria affecting decision points are determined.

Step 2: Creating the hierarchy

This step is the process of decomposing a decision problem into sub-problems in a hierarchical order that will make it easier to grasp and evaluate of problem. The hierarchy structure of AHP is shown in Figure 5.

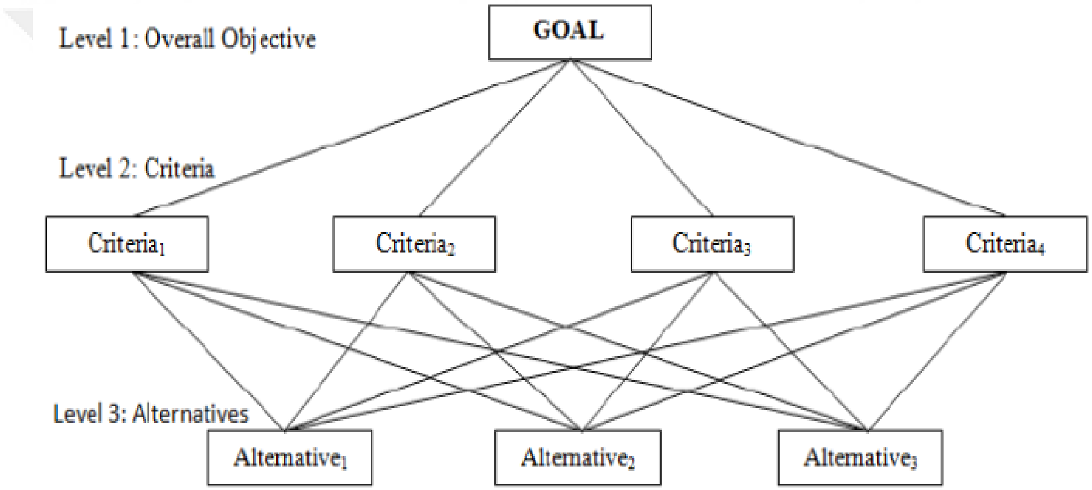


Figure 5 General Hierarchy Structure of AHP

Step 3: Creating a Comparison Matrix between Criteria

The comparison matrix between criteria is an nxn square matrix. The matrix components on the diagonal of this matrix take the value 1. The comparison matrix is shown below.

$$\text{Alternatives} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Components on the diagonal of the comparison matrix, i.e. in the case of i = j, take the value 1. Because in this case, the relevant factor is compared with itself. Comparisons of criteria are made one-to-one and mutually in line with their relative importance values. Score scale should be used which is proposed at Table 3 by Saaty when these matrices are forming [53].

Table 3 Scoring Scale of AHP

Intensity of Importance	Definition	Explanation
1	Equal Importance	Requirements i and j are equal value
3	Moderate Importance	Requirements i has slightly higher value than j
5	Strong Importance	Requirements i has strongly higher value than j
7	Very Strong Importance	Requirements i has very strongly higher value than j
9	Extreme Importance	Requirements i has very an absolutely higher value than j
2,4,6,8	Intermediate Values	These are Intermediate scales between two adjacent judgments

Comparisons are made for all values above the diagonal of 1 in the comparison matrix. Naturally, it will be sufficient to use the formula(0.1) below for the components below the diagonal.

$$a_{ji} = \frac{1}{a_{ij}} \tag{0.1}$$

Step 4: Determining the Importance value of the Criteria

The comparison matrix shows the importance levels of the criteria relative to each other within a certain logic. Column vectors forming the comparison matrix are used to determine percentage significance distributions. B column vector with n grain and n components is constructed.

$$B_i = \begin{bmatrix} b_{11} \\ b_{21} \\ . \\ . \\ . \\ b_{n1} \end{bmatrix} \tag{0.2}$$

The formula below is used to calculate the B column vectors.

$$b_{ij} = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}} \tag{0.3}$$

When the steps described above are repeated for all criteria, B column vectors will be obtained as many as the number of factors. When n grain B column vectors are combined in a matrix format, the C matrix shown below will be created.

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ . & & & . \\ . & & & . \\ . & & & . \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} \quad (0.4)$$

Using the C matrix, the arithmetic average of the row components forming the C matrix is obtained as shown in the formula and the column vector W called the Priority Vector is obtained.

$$w_i = \frac{\sum_{j=1}^{n} c_{ij}}{n} \quad (0.5)$$

Percentage of importance level of criteria can be presented by

$$W = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ . \\ w_n \end{bmatrix} \quad (0.6)$$

Step 5: Measuring Consistency in Criterion Benchmarks

The realism of the results will depend on the consistency of the decision maker in the one-to-one comparison between the criteria. AHP proposes a process for measuring consistency in these comparisons. For the consistency index, λ must be calculated. For the calculation of λ, the D column vector is obtained from the matrix product of the comparison matrix A and the W priority vector.

$$D = [aij]nxn \text{ x } [wi]n \text{ x1} = [di] \text{ nx1} \quad (0.7)$$

$$\lambda = \frac{\sum_{i=1}^{n} E_i}{n} \tag{0.8}$$

After calculating a, the Consistency Indicator (CI) can be calculated using the formula below.

$$CI = \frac{\lambda - n}{n - 1} \tag{0.9}$$

In the last stage, CI is divided by the standard correction value called Random Indicator (RI) and shown in Table 4 and CR is obtained with the following formula.

$$CR = \frac{CI}{RI} \tag{0.10}$$

Table 4 Random Consistency Index Table

N	**RI**
1	0
2	0
3	0,58
4	0,90
5	1,12
6	1,24

If the calculated CR value is less than 0.10, it shows that the comparisons made by the decision maker are consistent. A CR value greater than 0.10 indicates either a calculation error in AHP or the inconsistency of the decision maker in comparisons.

Platform: Super Decisions software version 2.10.0 was used to implement the AHP method in this study. The software provides to calculate weights and compare pairwise alternatives and criteria

CHAPTER 4

RESULTS AND FINDINGS

An industrial case study has been conducted in this thesis for the purpose of selection of continuous integration tool that meets the criteria and features suitable for a closed networked institution. Results gathered from the study will be explained in the following sections.

4.1. Study Group

The study group is composed of 41 software professionals. Distribution chart of the people according to the department they graduated from is given in Figure 6. The number of electronic engineers is high due to the predominantly embedded software development projects in the institution. 25 (61%) of the engineers were electronic engineers and, 16 (39%) of them were computer engineers.

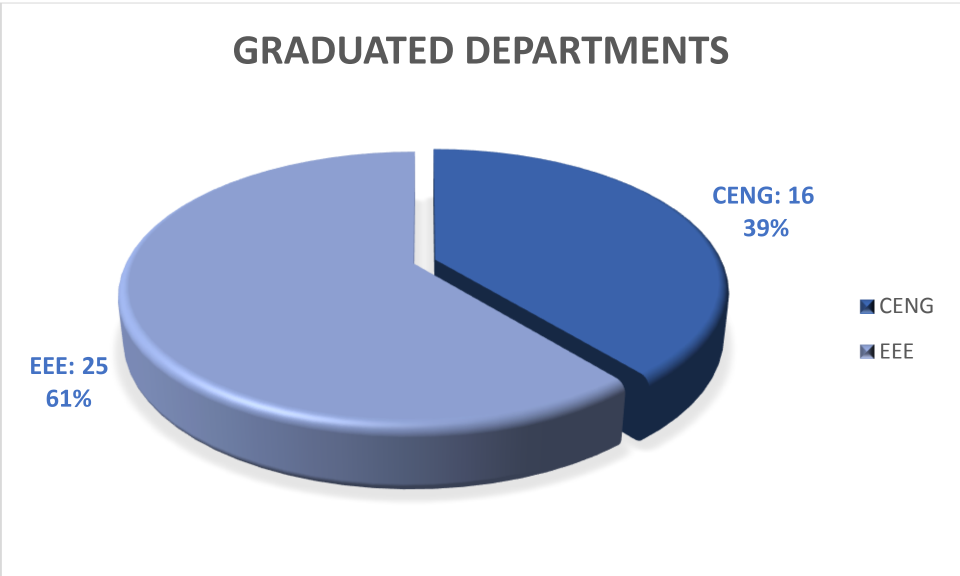


Figure 6 Graduated Department

The unit where participants work is shown in Figure 7. 18 (%44) of them were from embedded software system unit. Then comes the software testing unit where 9 (22%) engineers work. A software simulation team with 8 (19%) people follows them, and finally, there was a software architecture team of 6 (15%) people.

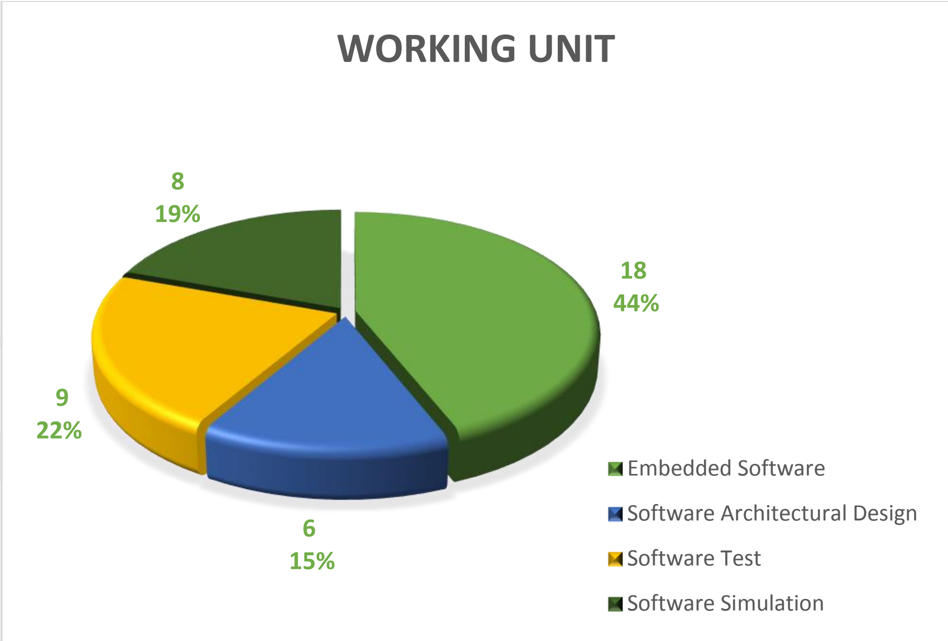


Figure 7 Working Unit Distribution

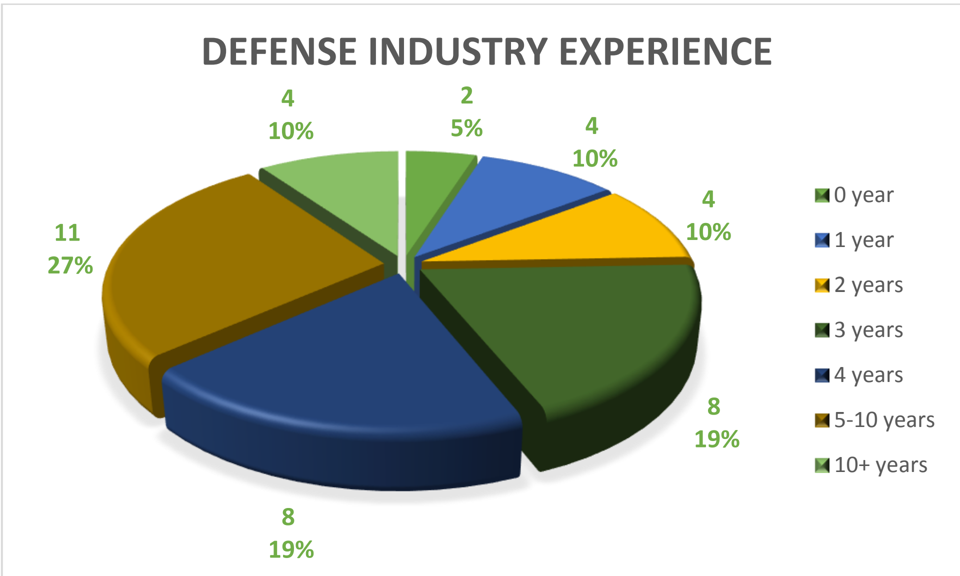


Figure 8 Defense Industry Experience

The experience of the working group in the defense industry is shown in Figure 8. Those with more than 5 years of experience make up 15 (37%) of the group, and those with less than 5 years of experience make up 26 (63%). At the same time, Figure 9 shows the experiences of the participants outside the defense industry.16 (39%) of them have no experience in other sectors.

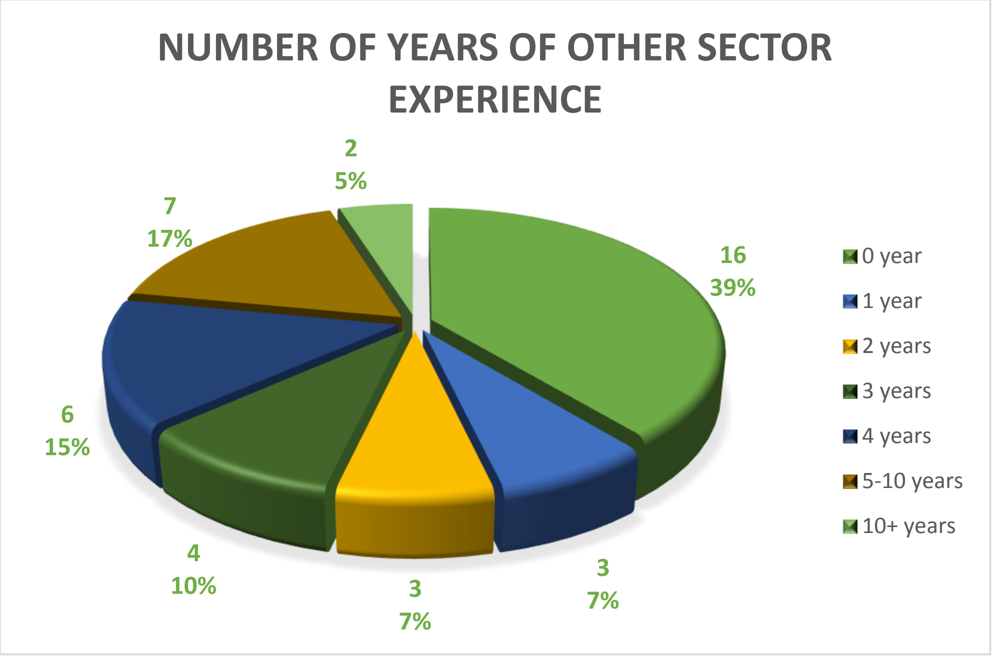


Figure 9 Other Sector Experience

Additionally, 22 (54%) of the participants were studying for their master's degree (Figure 10).

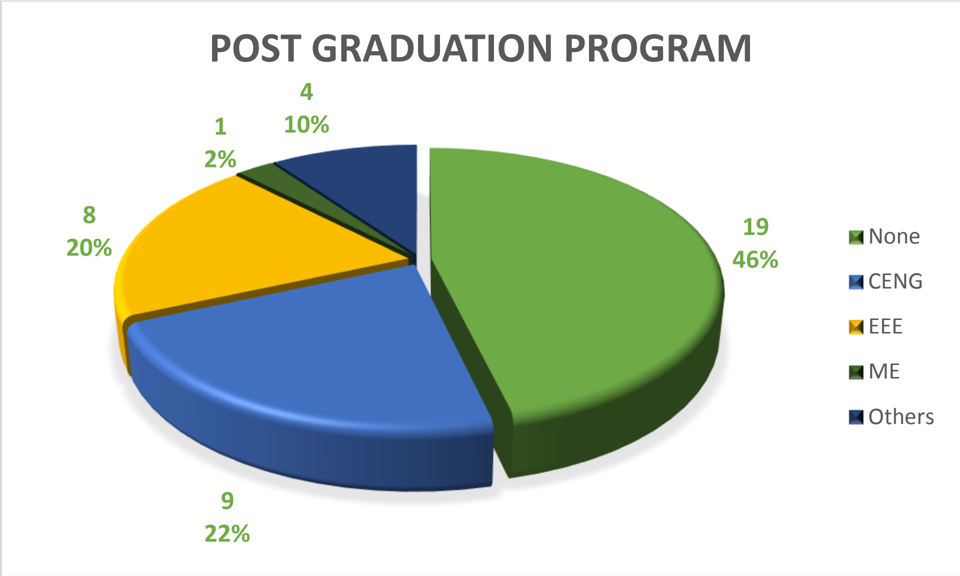


Figure 10 Post Graduation Program

The study group was asked about their previous experiences of continuous integration. As we can see in Figure 11, 17 (41%) of them had previous CI experience.

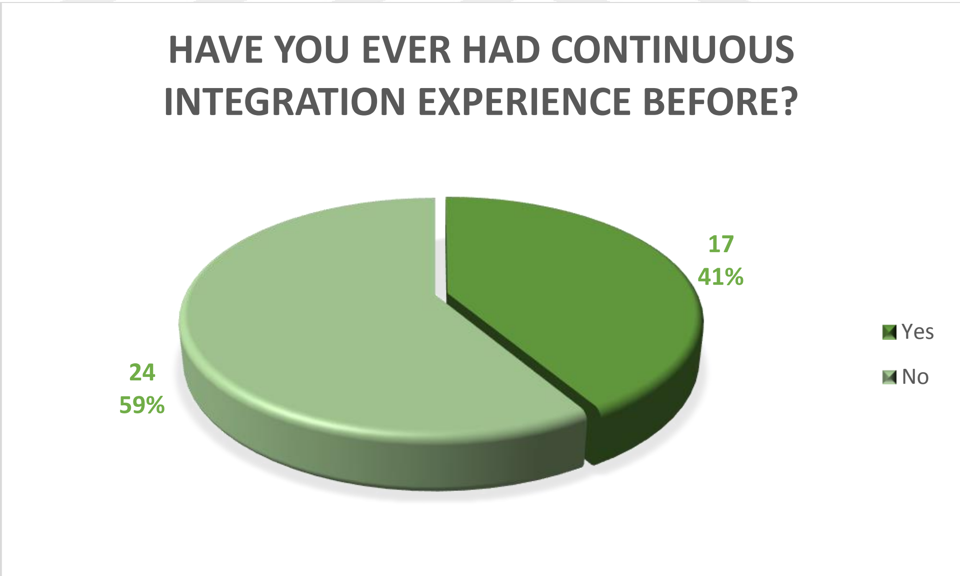


Figure 11 CI Experience Before

4.2. Expert Group

As seen in Table 5, the expert group consists of unit chiefs of 4 units and their coordinators.

Table 5 Information of Expert Group

Experts	Roles and Responsibilities	Year of Experience
Expert 1	Coordinator	14
Expert 2	Software Architecture Design Unit Chief	16
Expert 3	Embedded System Unit Chief	8
Expert 4	Software Test Unit Chief	5
Expert 5	Software Simulation and Mission Planning Unit Chief	6

4.3. Result of Expectations of Study Group from the Continuous Integration Tool Features

The study group was asked to choose their expectations from continuous integration tools in the questionnaire.

Figure 12 shows, the answers of the study group in the order from the feature they determined as most necessary to unnecessary. In the figure, it can be seen how many of the very necessary, necessary, neutral, not so necessary, not at all necessary options were preferred.

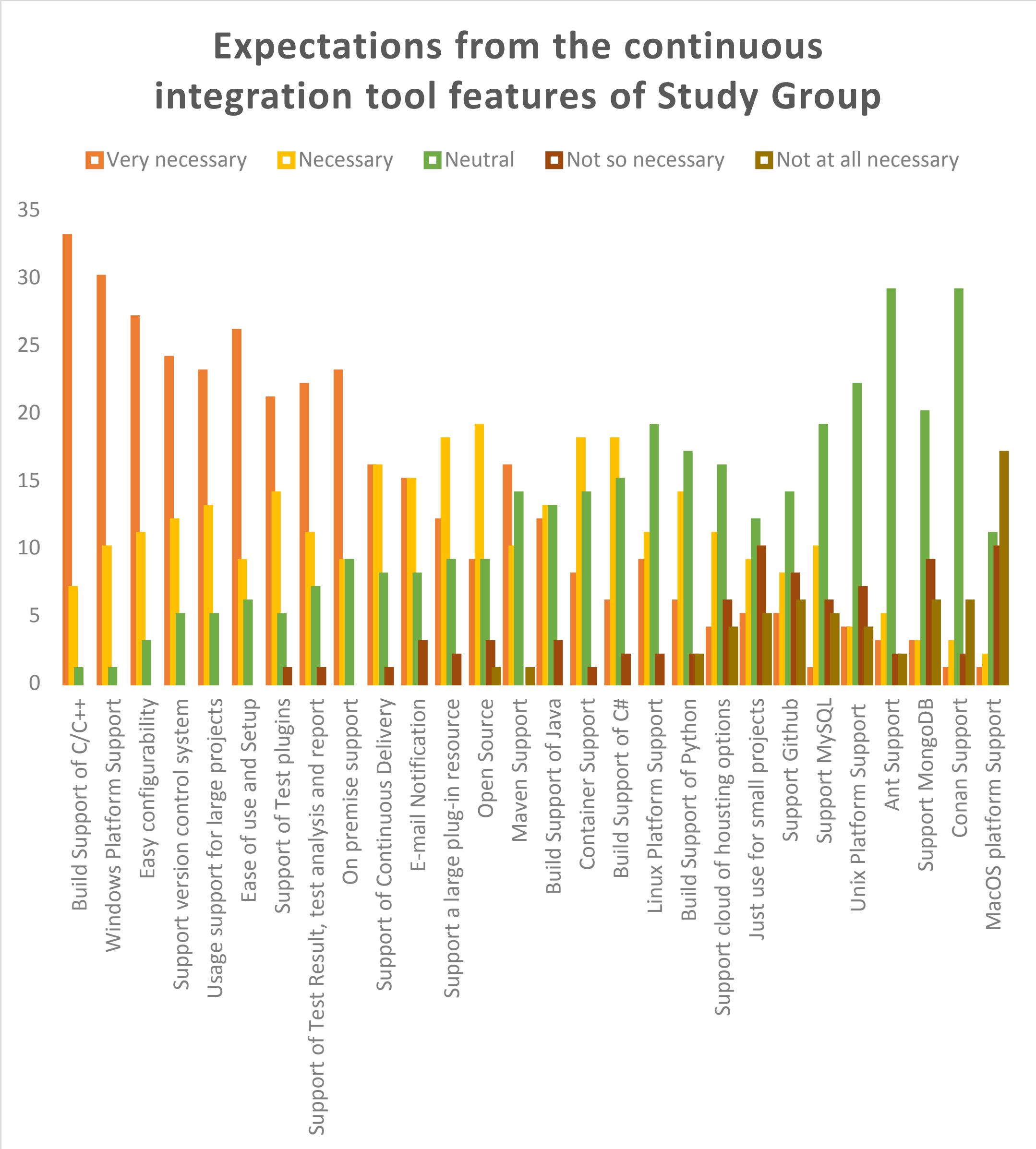


Figure 12 Expectations from the Continuous Integration Tool Features of Study Group

5-point Likert scale values were given for the options. Figure 13 shows the order of the results according to these values.

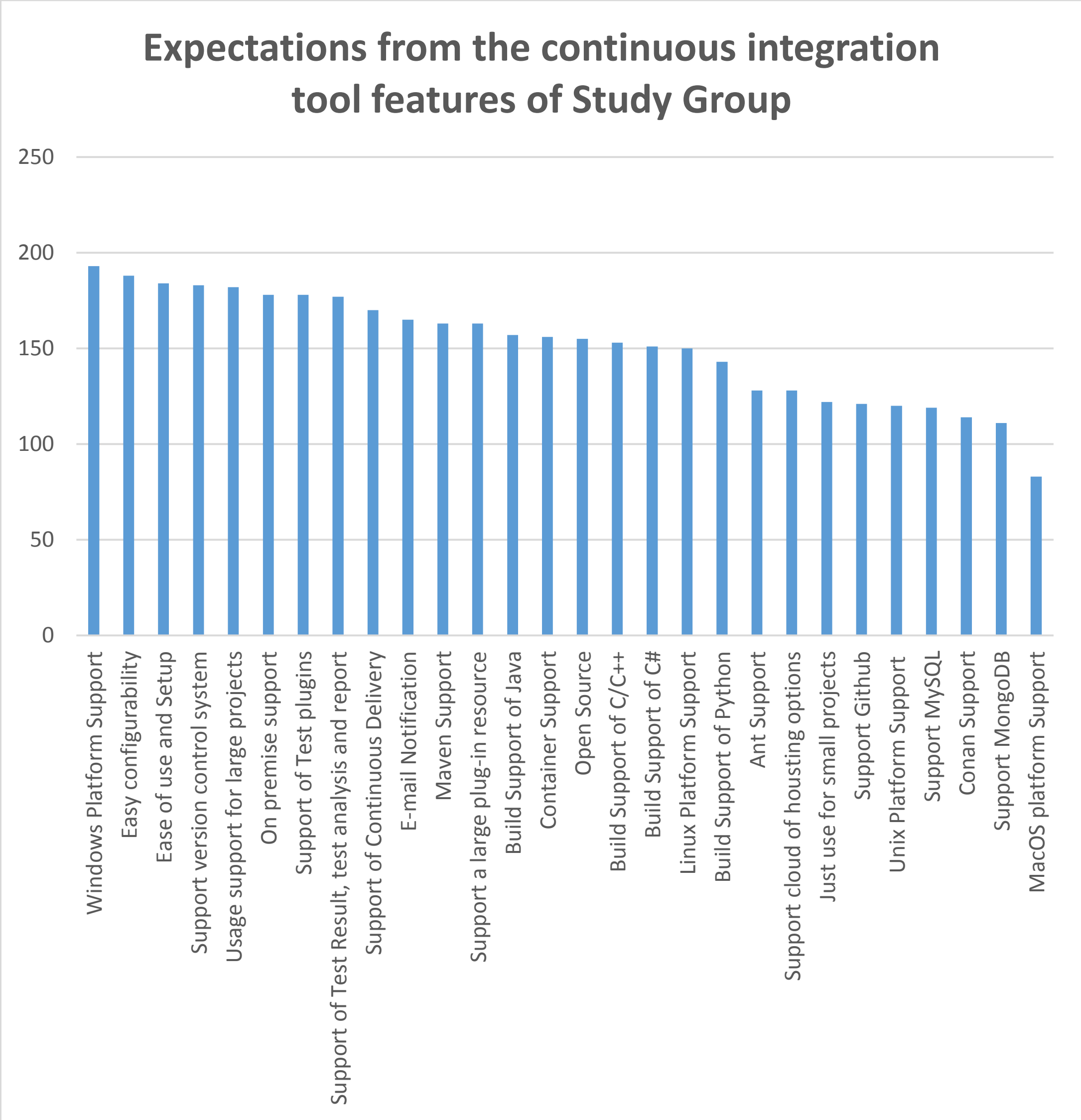


Figure 13 Expectations from the Continuous Integration Tool Features of Study Group

In Figure 14, only the sums of very necessary and necessary values (5,4) are listed. All these figures give us enough information about institution culture and the expectations of the participants from the CI tool.

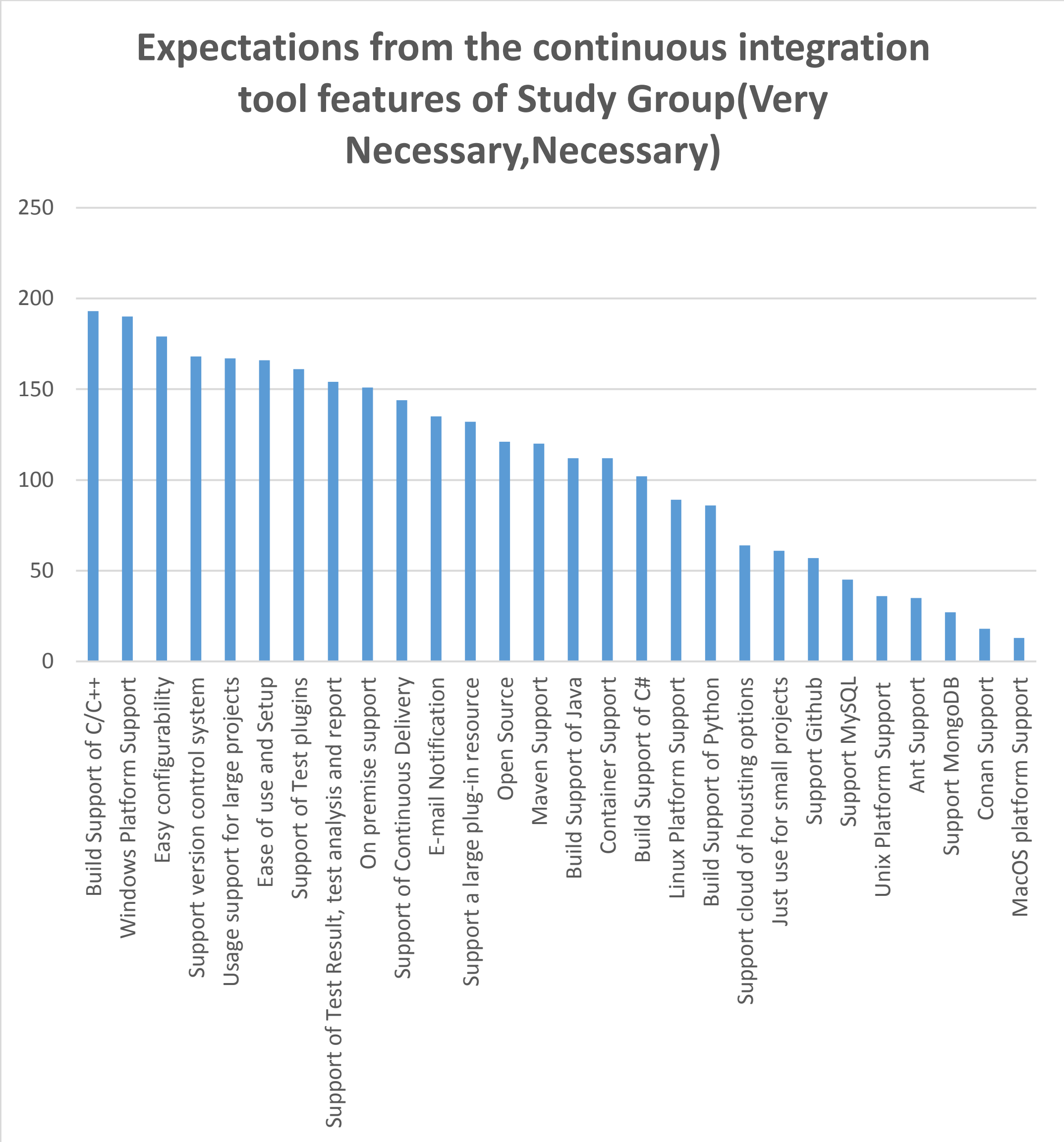


Figure 14 Expectations from the Continuous Integration Tool Features of Study Group (Very Necessary, Necessary)

The results show that the institution uses C / C ++ languages mostly. In addition, Windows has been the prominent choice as the Operating system. Version control has an important place since large projects are also worked on. In addition, we see that configurability ranks at the beginning in requests. It is also very important to have enough plugins in the software testing phase of the projects and to report the results. As an institution with a closed network, on premise support is among the required features. In addition, we see that CD, large plugin support, open source and Maven as a build automation tool are among the desired features for the company.

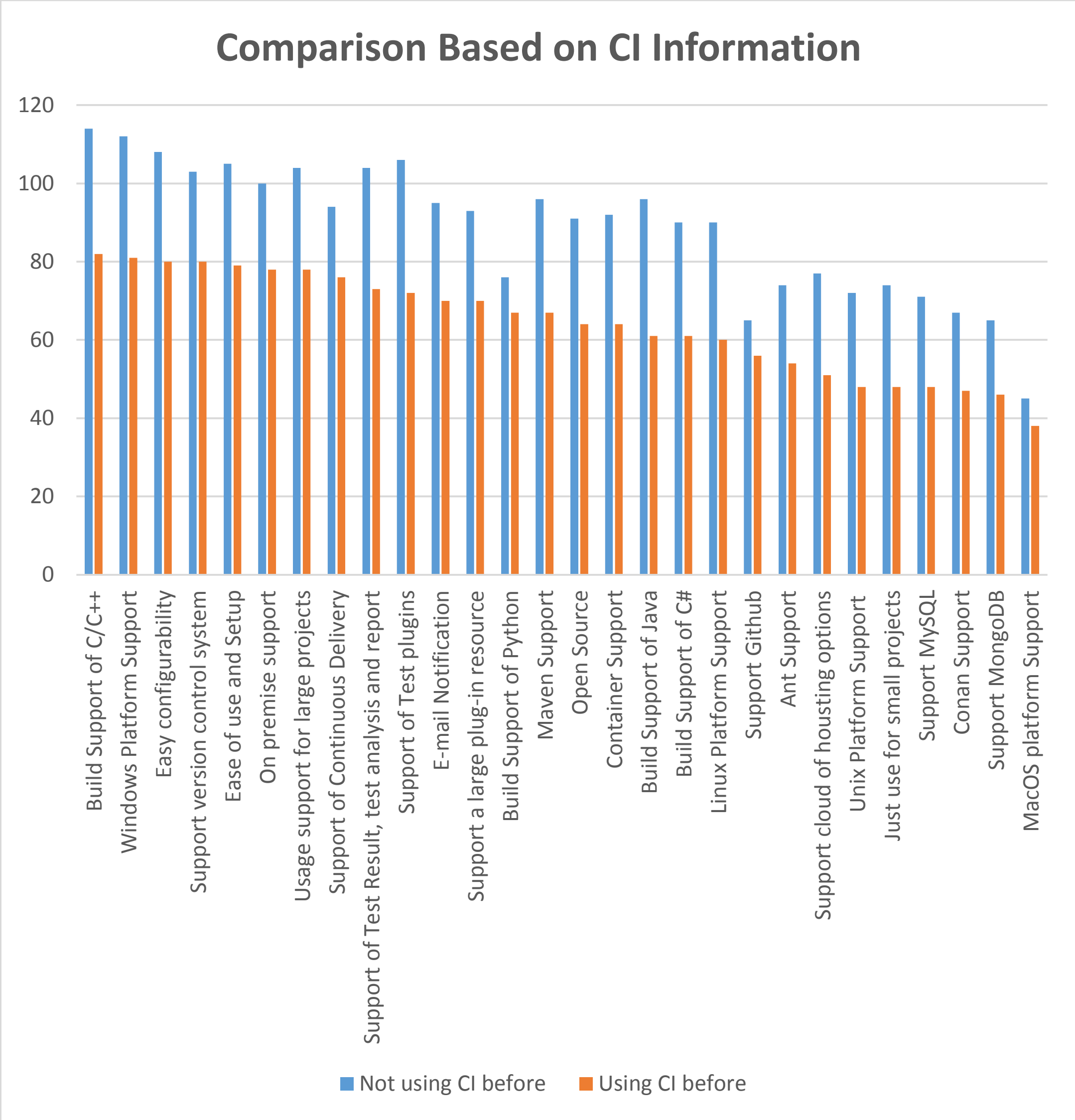


Figure 15 Comparison based on CI Information

The answers of the participants with previous CI experience and those without CI experience are given in Figure 15. This figure is ranked according to the answers of those who know the CI. It is noticed that there are some changes when comparing this ranking with the general ranking. It is seen that some features such as on premise, continuous delivery, ease of use and setup are brought forward according to the general order.

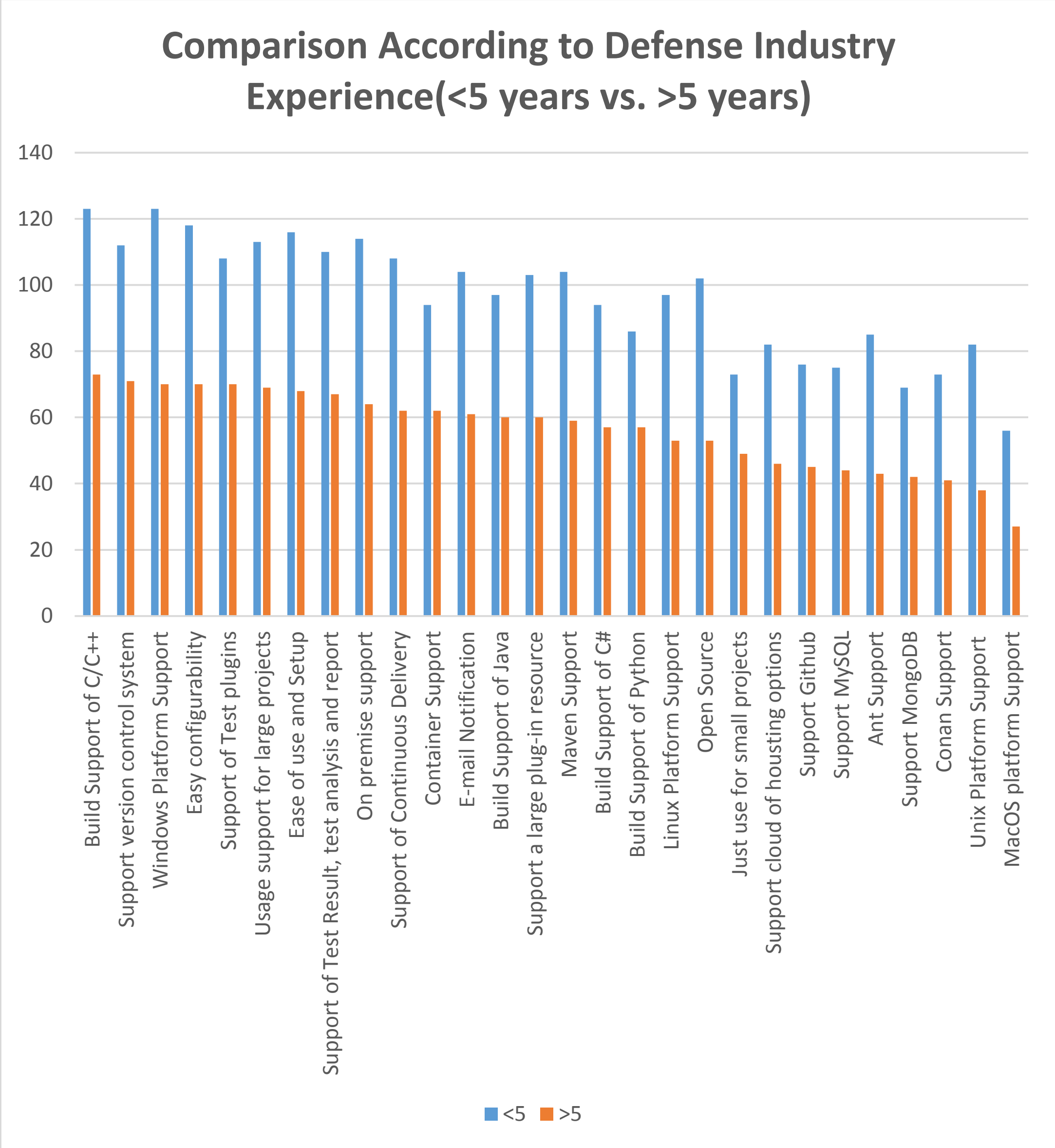


Figure 16 Comparison According to Defense Industry Experience (<5 year vs. >5 year)

Figure 16 shows how the answers of those with more than 5 years of experience and those with less than 5 years of experience rank. While ranking, the answers of the participants with more than 5 years of experience are considered. It is seen that the order in this figure is different when compared to the general ranking. However, it is striking that the listed values are very close to each other. Therefore, the difference in ranking is negligible.

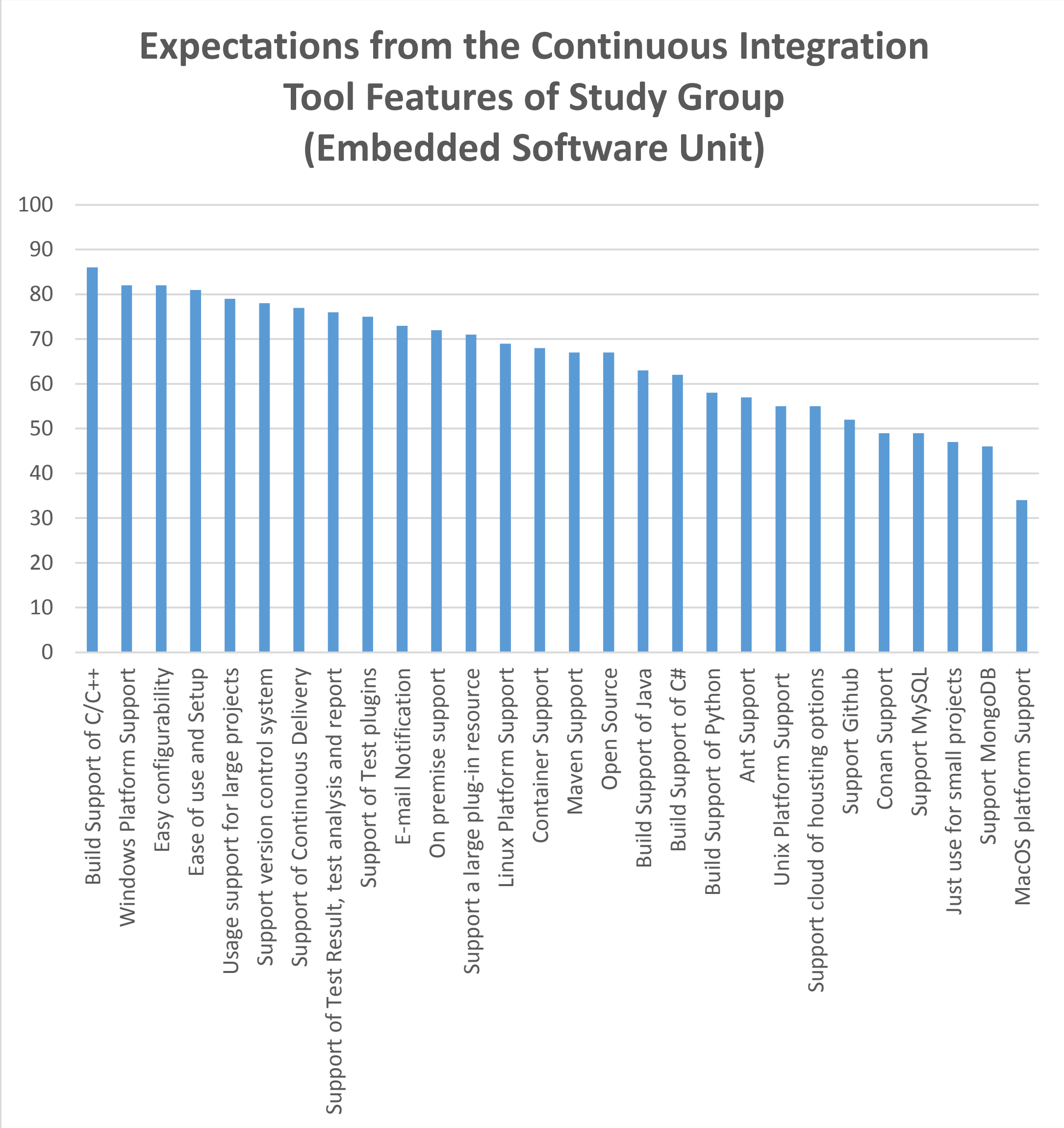


Figure 17 Expectations from the Continuous Integration Tool Features of Study Group (Embedded Software Unit)

The selection results of the embedded software unit, which constitutes the majority of the study group participating in the survey, is shown in Figure 17. While the platform and language features, which are the first two features in the general ranking keep their places, the ranking changes with slight score differences are noticed in the other feature rankings.

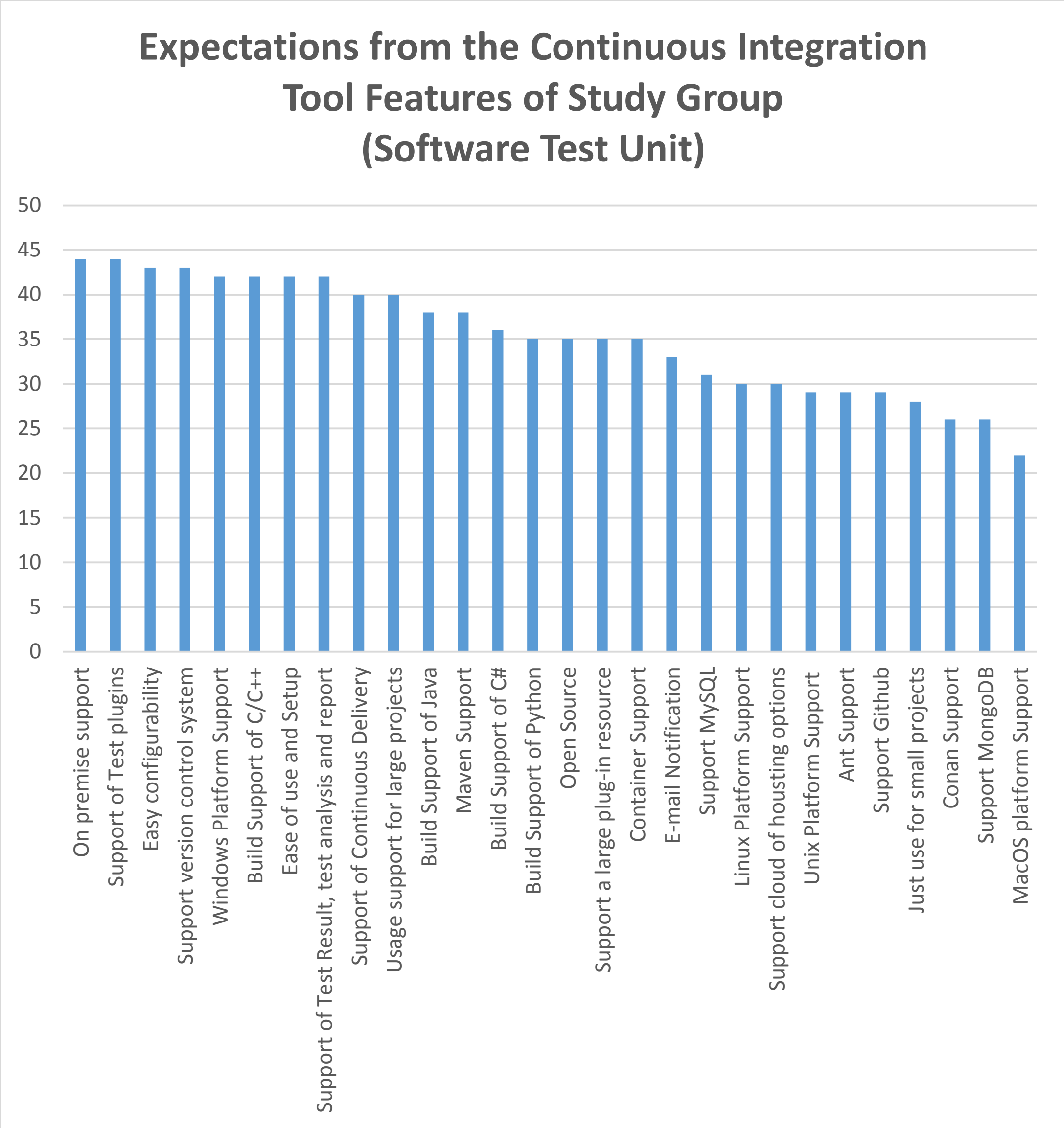


Figure 18 Expectations from the Continuous Integration Tool Features of Study Group (Software Test Unit)

The selection results of the software test unit, which is the second unit of the study group participating in the survey, is given in Figure 18. In the results of the test unit, we observe that the first-order properties have changed. On premise, test plugin and ease configurability features have overtaken platform and language features.

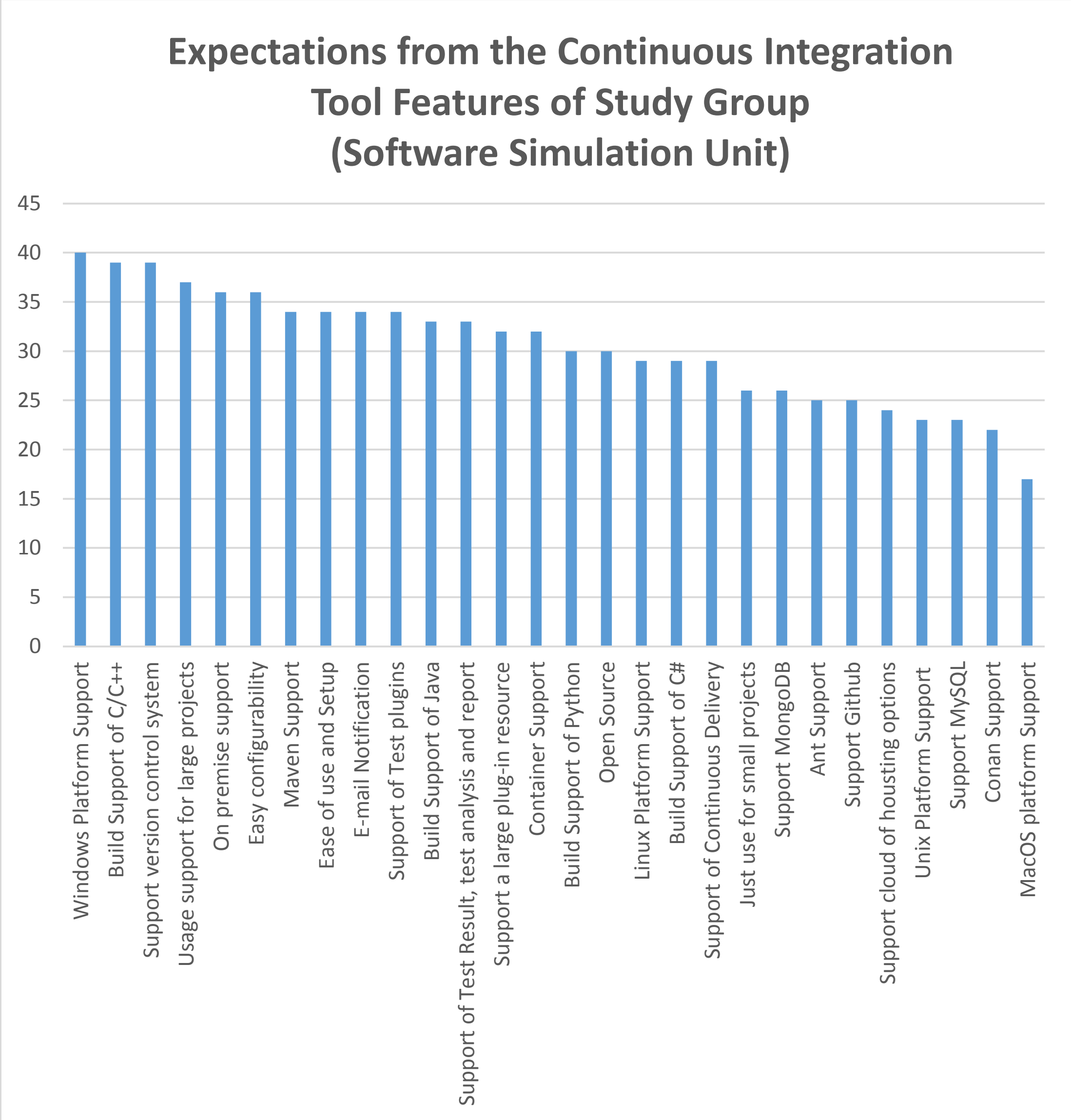


Figure 19 Expectations from the Continuous Integration Tool Features of Study Group (Software Simulation Unit)

The decisions the software simulation unit, which is the other unit of the study group participating in the survey, is shown in Figure 19.

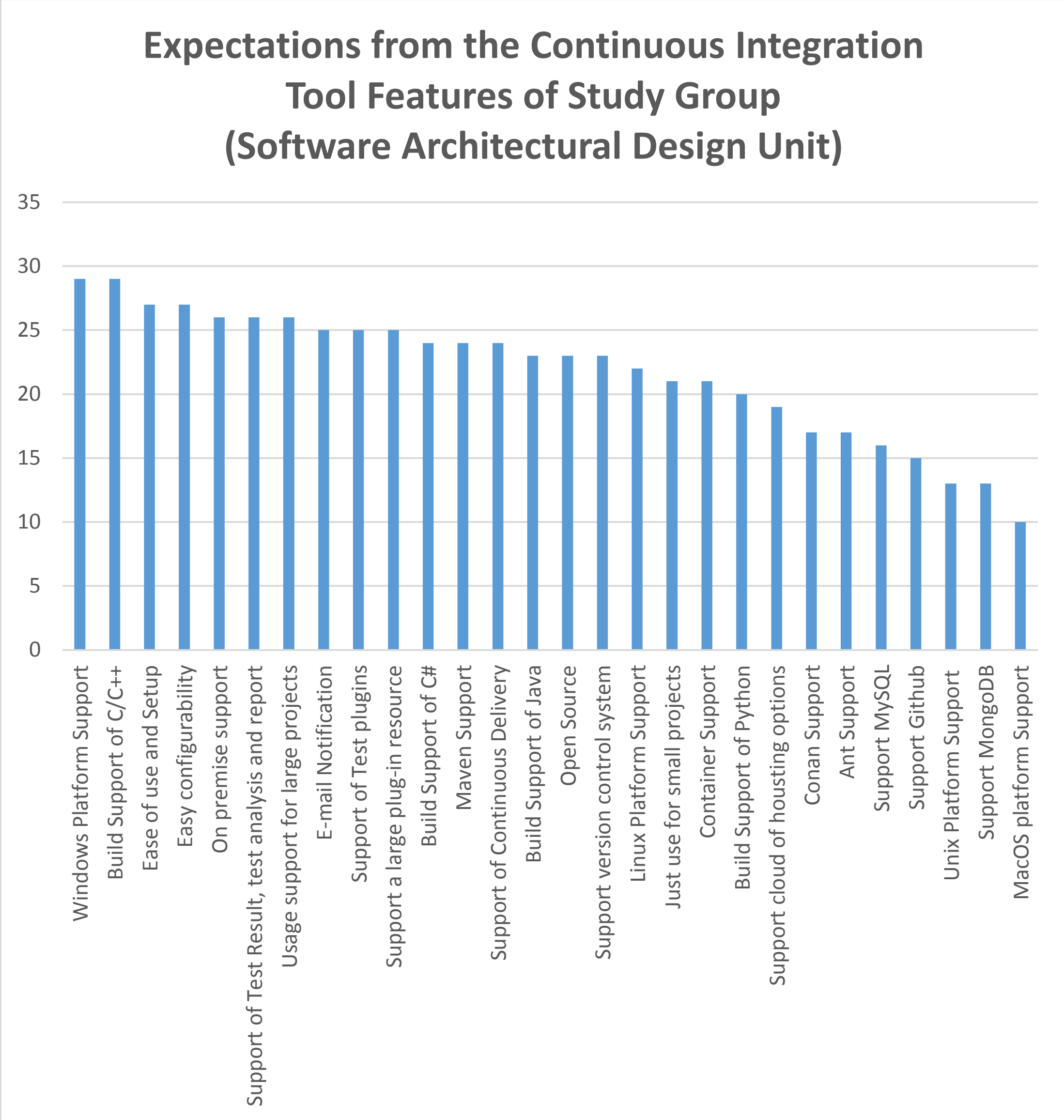


Figure 20 Expectations from the Continuous Integration Tool Features of Study Group (Software Architectural Design Unit)

The decision of the software architectural design unit, which is the last unit of the study group participating in the survey, is shown in Figure 20.

4.4. Determination of Alternatives

As TÜBİTAK SAGE software team, it has been decided to use continuous integration in terms of build automation, version controls and tests automation. Since it is difficult to switch to this structure and set an own repository in a closed network, it is necessary to choose the right tool. It has been noticed that there are more than 51 CI tools on the market. Some of these pieces have been eliminated because they have primitive properties. The needs of the team that will use the tool were determined through the survey we made for the study group. Expert group evaluated the tools in the market in line with these needs. As a result of the evaluation of the expert group, the points in the market and user evaluations, 4 most suitable alternatives from 35 tools were determined.

Table 6 The CI Tool Alternatives

CI Tool Alternatives	Tool Name
Alternative 1	Jenkins
Alternative 2	TeamCity
Alternative 3	Bamboo
Alternative 4	Buddy

4.5. Determination and Reduction of Criteria

Afterwards, the expert group evaluated the results resulting from the survey of the study group. As a result of this evaluation, we can see the criteria chosen by the expert group in Table 7.

Table 7 Selected Criteria of Expert Group

Expert Group	Selected Criteria
Expert 1	Functionality Flexibility/Expandability Compatibility Usability/Availability
Expert 2	Functionality Flexibility/Expandability Compatibility Reliability
Expert 3	Functionality Flexibility/Expandability Compatibility Usability/Availability Reliability
Expert 4	Functionality Flexibility/Expandability Compatibility Usability/Availability Reliability Longevity
Expert 5	Functionality Compatibility Reliability

The selections for criteria made by the expert group are shown in Table 8. In this table, the frequencies of the choices made can also be seen. The criteria selected by at least 4 experts are in green color, the others are in red color. The AHP method criteria group consists of those in green color.

Table 8 Selection Frequency of Criteria

Criteria	Selection Frequency
Functionality	5
FlexibiIity&Expandability	4
Compatibility	5
Usability&Availability	3
Reliability	5
Longevity	1
Open Source & Commercial	0

To sum up, the criteria set that the expert group wants to be in the continuous integration tool are shown in Table 9.

Table 9 Determined Criteria

The Criteria
Functionality
FlexibiIity&Expandability
Compatibility
Reliability

4.6. Implementation of AHP

4.6.1. Solution of the Problem

In Figure 21, AHP hierarchical structure, which is the selection of continuous integration tool suitable for corporate culture, is shown.

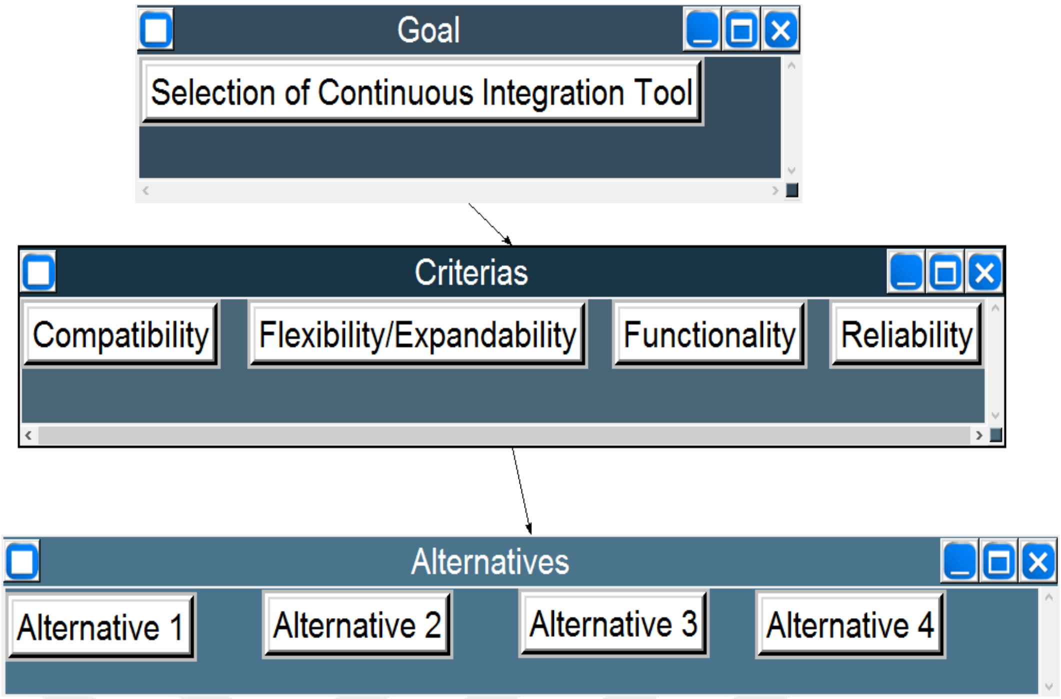


Figure 21 AHP Hierarchy Structure of the Component Selection

4.6.2. Determination of Weighting Criteria

The expert group weighed the criteria by pairwise comparisons. All comparison steps are included in the Appendix.

The result of Expert 1's evaluation between criteria is shown in Figure 22. It appears that the highest value criterion with 0.50136 is reliability. The criterion with the lowest value with 0.046 is flexibility & expandability

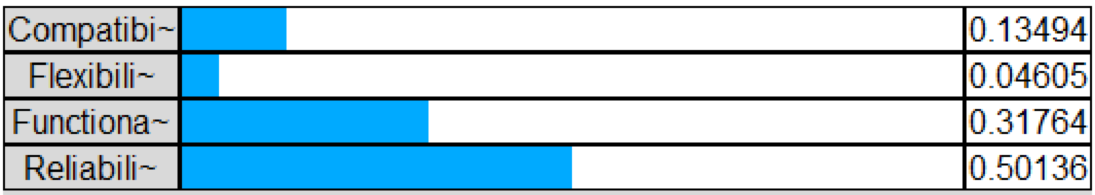


Figure 22 The Weight of Criteria by Expert 1

The result of Expert 2's evaluation between criteria is given in Figure 23. It appears that the highest value criterion is functionality with 0.29686. The criterion with the lowest value is compatibility with 0.19794.

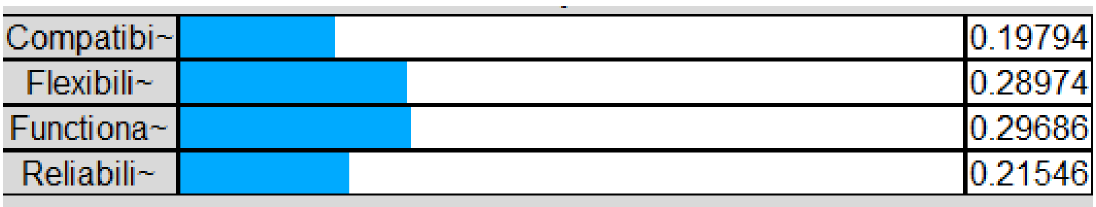


Figure 23 The Weight of Criteria by Expert 2

The result of Expert 3's evaluation between criteria is shown in Figure 24. It appears that the highest values criterion are compatibility and functionality with 0.30500. The criterion with the lowest value is flexibility & expandability with 0.11314.

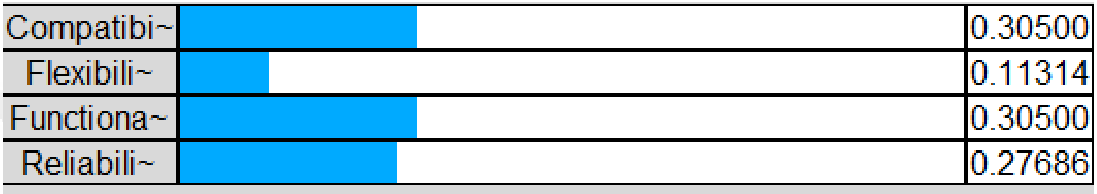


Figure 24 The Weight of Criteria by Expert 3

The result of Expert 4's evaluation between criteria is shown in Figure 25. It appears that the highest value criterion is flexibility & expandability with 0.59869. The criterion with the lowest value is reliability with 0.07757.

Figure 25 The Weights of Criteria by Expert 4

The result of Expert 5's evaluation between criteria is given in Figure 26. It appears that the highest value criterion is functionality with 0.64793. The criterion with the lowest value is flexibility & expandability with 0.06236.

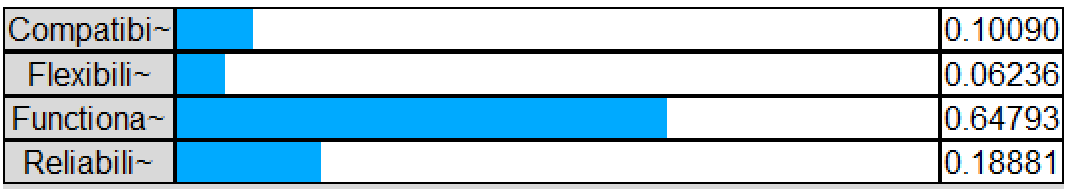


Figure 26 The Weights of Criteria by Expert 5

Figure 27 was shown the results that the average of all values.

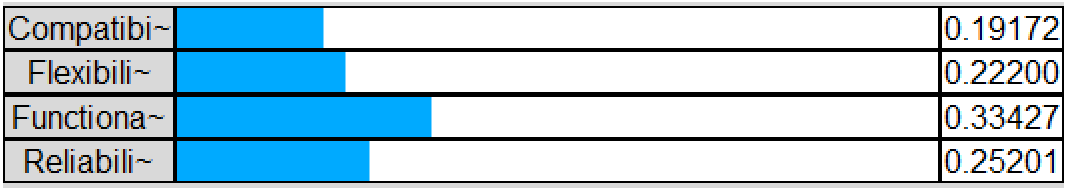


Figure 27 The Means of Criteria Weights

As a result of the decision matrix created with the Super Decision program, the weights of the criteria were obtained. The priorities for the criteria set for the continuous integration tool selection problem is in Figure 27. According to this figure, the Functionality criterion has the maximum weight of 0.33427. The compatibility criterions followed by 0.19172.

4.6.3. Evaluation of Alternatives

The following information is given as a file for the expert group to evaluate the alternatives.

- Explanations of selected criteria Table 2.
- Information about the Saaty scale, which is the scale of the AHP method Table 3.
- Link of general information about alternatives.
- Link of detailed technical information about alternatives.
- Link of alternatives in the market place. These websites contain user comments, user comments, and user ratings.
- Links of the alternative usage as a video demo.

Before the expert group made a pairwise comparison between the alternatives, all the above documents were given to the expert group as a file. Based on given information, they made expert group evaluations and returned the results.

After evaluating the criteria with the expert group, they evaluated alternative CI tools among themselves. According to the average weight values of the criteria determined by the expert group, the evaluation results of the alternatives are given in the tables below.

From Figure31-Figure35 we can see the individual results of the expert group.

Alternatives	Normalized By Cluster
Alternative 1	0.360715
Alternative 2	0.217901
Alternative 3	0.284582
Alternative 4	0.136803

Figure 28 Priorities of the Alternatives by Expert 1

Alternatives	Normalized By Cluster
Alternative 1	0.520478
Alternative 2	0.267247
Alternative 3	0.104524
Alternative 4	0.107752

Figure 29 Priorities of the Alternatives by Expert 2

Alternatives	Normalized By Cluster
Alternative 1	0.310296
Alternative 2	0.174736
Alternative 3	0.250107
Alternative 4	0.264860

Figure 30 Priorities of the Alternatives by Expert 3

Alternatives	Normalized By Cluster
Alternative 1	0.357487
Alternative 2	0.269989
Alternative 3	0.207848
Alternative 4	0.164677

Figure 31 Priorities of the Alternatives by Expert 4

Alternatives	Normalized By Cluster
Alternative 1	0.308184
Alternative 2	0.513217
Alternative 3	0.108910
Alternative 4	0.069689

Figure 32 Priorities of the Alternatives by Expert 5

When the priorities are averaged as a result of the evaluation of each expert in the tables above, the final priorities is obtained as given in Table 10 below.

Table 10 Mean Priorities of the Alternatives

Mean Priorities of the Alternatives	**Normalized By Cluster**
Alternative 1	0,371432
Alternative 2	0,288618
Alternative 3	0,1911942
Alternative 4	0,1487562

All these operations and the creation of the decision matrix were done with the Super Decision application. As a result of the calculations, the CI tool alternative 1 has highest value with 0.371432 and CI tool alternative 4 has lowest value with 0.1487562.

4.6.4. Limitation

In this study, the completion of the questionnaire, the evaluation of the results, and the application of the AHP method to the expert group were completed in approximately 1-month period. In other words, 1 month was spent to make the right tool selection for the institution. Some commercial companies may not be able to devote that much time to select a CI tool. In such cases, it will be possible to shorten the duration of tool selection by shortening the time spent for the survey and by applying AHP method with less experts.

4.6.5. Discussion

With the AHP method applied, the continuous integration tool that is the most suitable for institution's culture and working style was determined. In the selection method, the result was obtained with the participation of 5 experts in the field. Firstly, each member of the expert group was asked independently from each other to evaluate the criteria determined by pairwise comparisons. As a result of this evaluation, average criterion weight values were obtained. Technical informations, market reviews and a demo of all continuous integration alternatives were given to the expert group for their review. In the light of this information, they were asked to make a pairwise comparison of alternatives among themselves. As a result of these evaluations, the priorities values as a final result was obtained by using average criteria weight values.

In a similar study in this area, Polkhovskiy tried to make a suitable choice by comparing the CI tool features in the market [15]. In his study, Jenkins was chosen as it is very suitable for starting and has many features thanks to its many plugins. Also, in his study, Polkhovskiy, mentions that the selection is very difficult due to the fact that the selection includes many criteria and there are too many tools, and it is even indecisive between a few [15]. Polkovsky has included seven criteria in his study. These are functionality, compatibility, reliability, longevity, usability, free, open-source or commercial whereas in this study, flexibility, expandability, availability, are also considered as extra criteria [15].

On the other hand, Makam et al. evaluated the CI tools on cloud versus on premise in terms of price, flexibility, security, and the reliability criteria while comparing [25].

Lerra et al. evaluated CI tools in terms of monitoring application performance. Seven criteria were evaluated in this study. These are availability, maintenance, support, flexibility, portability, deployment, suitability. This evaluation has been made from a specific point of view about performance monitoring, no general evaluation has been made [29].

In this study, difficulty of CI tool selection is reduced by the application of the multi-scale decision making method, AHP.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, the appropriate continuous integration tool selection is made using a systematic process and multi-scale selection method for a company that has a closed network. Continuous integration, one of today's technologies that is part of the agile methodology, has an important place for software projects. For institutions with closed networks, it is very important to choose the appropriate tool when transitioning to this type of new technology. Choosing the wrong tools brings financial and moral damage to the institution.In line with all these, the most suitable continuous integration tool for the institution culture was selected as a result of the evaluation of the expert group with the requests determined with the participation of the study group.

In this case study, the survey results of the study group were evaluated by the expert group in AHP application process. As a result of this evaluation, functionality was considered by the expert group to be the most important criterion. Considering the importance of the projects, the reliability criterion has not been ignored. In addition, flexibility and expandability criteria were also preferred in line with the demand for different additions in the graphic results. It follows with a very close value in compliance criteria which was selected considering the priorities of the units.

As a result of evaluating the alternatives, the expert group ranked Alternative 1 first with a rate of 37.14%. During the evaluation of the alternatives, it was ensured that the experts were not affected by each other. In addition, sufficient time was given to appraise the alternatives. The expert group evaluated the alternatives among themselves in the light of this information and in line with the wishes of the working group. Comparing the alternatives with each other using the pairwise comparison method prevented them from focusing on a single alternative. The average of the five experts' evaluations made the study more objective. In addition, the CI tool selected in this study has been started to be used by TÜBİTAK SAGE.

In the litarature, to the best of our knowledge, there is no study that uses AHP method for the selection of CI tools. In addition, the modification of the AHP process applied is a part of the originality of the study. Indeed, AHP was applied to 5 experts. Also, the questionnaire conducted with the study group during the determination of alternatives and criteria distinguishes this study from other AHP studies.

This case study has proven that AHP method can be used efficiently in the selection of a continuous integration tool in the software field. This study has been an exemplary study showing that AHP methodology can be used in a situation that requires multiple decision-making methods in software field. With the right tool selection, productivity has increased and time has been saved. As a future study, this study can be supported with fuzzy AHP method, and hybrid method work can be achieved by using ANP and TOPSIS methods.

REFERENCES

[1] "It has been my observation that most people get ahead during the ... - Quote." [Online]. Available: https://quotes.yourdictionary.com/author/quote/574189. [Accessed: 06-Nov-2020].

[2] C. Jones, "Software Quality Metrics: Three Harmful Metrics and Two Helpful Metrics," *Namcook Anal. LLC*, 2012.

[3] C. Jones and O. Bonsignour, *The Economics of Software Quality*. 2012.

[4] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, pp. 246–270, 2020.

[5] D. Sunner, "Agile : Adapting to need of the hour," *2016 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol.*, pp. 130–135, 2016.

[6] K. Gaurav and B. K. Pradeep, "Impact of Agile Methodology on Software Development Process," *Int. J. Comput. Technol. Electron. Eng.*, vol. 2, no. 4, pp. 46–50, 2012.

[7] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "From Agile to DevOps: Smart Skills and Collaborations," *Inf. Syst. Front.*, vol. 22, no. 4, pp. 927–945, 2020.

[8] S. A. I. B. S. Arachchi and I. Perera, "Continuous integration and continuous delivery pipeline automation for agile software project management," *MERCon 2018 - 4th Int. Multidiscip. Moratuwa Eng. Res. Conf.*, pp. 156–161, 2018.

[9] B. Fitzgerald and K. J. Stol, "Continuous software engineering: A roadmap and agenda," *J. Syst. Softw.*, vol. 123, pp. 176–189, 2017.

[10] "Continuous Integration." [Online]. Available: https://martinfowler.com/articles/continuousIntegration.html. [Accessed: 08-Nov-2020].

[11] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017.

[12] A. Khan, H. Chawla, H. Kathuria, H. Chawla, and H. Kathuria, “Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps,” *Researchgate*, no. June, pp. 1–10, 2017.

[13] N. Seth and R. Khare, “ACI (automated Continuous Integration) using Jenkins: Key for successful embedded Software development,” *2015 2nd Int. Conf. Recent Adv. Eng. Comput. Sci. RAECS 2015*, no. December, 2016.

[14] S. Bobrovskis and A. Jurenoks, “A survey of continuous integration, continuous delivery and continuous deployment,” *CEUR Workshop Proc.*, vol. 2218, pp. 314–322, 2018.

[15] D. Polkhovskiy, “Denis polkhovskiy comparison between continuous integration tools,” *Indian J. Sci. Technol.*, vol. 7, no. April, 2016.

[16] G. P. White, “The implementation of management science in higher education administration,” *Omega*, vol. 15, no. 4, pp. 283–290, 1987.

[17] E. Triantaphyllou, “MCDM Methods: A Comparative Study,” 2000.

[18] “Top Continuous Integration Tools: 51 Tools to Streamline Your Development Process, Boost Quality, and Enhance Accuracy – Stackify.” [Online]. Available: https://stackify.com/top-continuous-integration-tools/. [Accessed: 02-Dec-2020].

[19] “Continuous Integration Tools Market Share, Size and Industry Growth Analysis 2020 - 2025.” [Online]. Available: https://www.industryarc.com/Research/Continuous-Integration-Tools-Market-Research-500805. [Accessed: 01-Dec-2020].

[20] E. Laukkanen, J. Itkonen, and C. Lassenius, “Problems, causes and solutions when adopting continuous delivery—A systematic literature review,” *Inf. Softw. Technol.*, vol. 82, pp. 55–79, 2017.

[21] A. Brunnert *et al.*, “Performance-oriented DevOps: A Research Agenda,” Aug. 2015.

[22] “Devops: A Software Revolution in the Making? | Cutter Consortium.” [Online]. Available: https://www.cutter.com/article/devops-software-revolution-making-416511. [Accessed: 27-Nov-2020].

[23] “Devopsdays Ghent 2009.” [Online]. Available: https://legacy.devopsdays.org/events/2009-ghent/. [Accessed: 27-Nov-2020].

[24] S. A. I. B. S. Arachchi and I. Perera, “Continuous integration and continuous

delivery pipeline automation for agile software project management," *MERCon 2018 - 4th Int. Multidiscip. Moratuwa Eng. Res. Conf.*, no. July, pp. 156–161, 2018.

[25] V. Makam and A. Express, "Continuous Integration on Cloud Versus on Premise : A Review of Integration Continuous Integration on Cloud Versus on Premise : A Review of Integration Tools," no. June, 2020.

[26] "10 Things to Consider While Choosing a CI Platform - DZone DevOps." [Online]. Available: https://dzone.com/articles/10-things-to-consider-while-choosing-a-ci-platform. [Accessed: 25-Nov-2020].

[27] R. Owen Rogers, "Scaling continuous integration," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3092, pp. 68–76, 2004.

[28] M. Brandtner, E. Giger, and H. Gall, "Supporting continuous integration by mashing-up software quality information," *2014 Softw. Evol. Week - IEEE Conf. Softw. Maintenance, Reengineering, Reverse Eng. CSMR-WCRE 2014 - Proc.*, pp. 184–193, 2014.

[29] V. G. V. Larrea, W. Joubert, and C. Fuson, "Use of Continuous Integration Tools for Application Performance Monitoring," pp. 1–12, 2015.

[30] M. Dağdeviren, "Decision making in equipment selection: An integrated approach with AHP and PROMETHEE," *J. Intell. Manuf.*, vol. 19, no. 4, pp. 397–406, 2008.

[31] N. Ahmad and P. A. Laplante, "Software project management tools: Making a practical decision using AHP," *Proc. 30th Annu. IEEE/NASA Softw. Eng. Work. SEW-30*, vol. 30, pp. 76–82, 2006.

[32] T. L. Saaty, "Decision making with the Analytic Hierarchy Process," *Sci. Iran.*, vol. 9, no. 3, pp. 215–229, 2002.

[33] B. Ö. Yilmaz, "Multi-Criteria Decision Making (MCDM) Applications in Military Healthcare Field," vol. 2, no. 2, pp. 149–181, 2020.

[34] H. Li, M. Lu, and Q. Li, "Software reliability metrics selecting method based on analytic hierarchy process," *Proc. - Int. Conf. Qual. Softw.*, pp. 337–344, 2006.

[35] M. N. Tekin, "SELECTION OF THE SOFTWARE DEVELOPMENT PROCESS MEASUREMENT COMPONENT ON SCRUM SOFTWARE

DEVELOPMENT: AN ANALYTIC HIERARCHY PROCESS APPROACH," 2019.

[36] M. C. Lin, C. C. Wang, M. S. Chen, and C. A. Chang, "Using AHP and TOPSIS approaches in customer-driven product design process," *Comput. Ind.*, vol. 59, no. 1, pp. 17–31, 2008.

[37] "20 Best Continuous Integration(CI) Tools in 2020." [Online]. Available: https://www.guru99.com/top-20-continuous-integration-tools.html. [Accessed: 23-Dec-2020].

[38] "Buildbot." [Online]. Available: https://buildbot.net/. [Accessed: 23-Dec-2020].

[39] "Bulut Bilişim Hizmetleri | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/tr-tr/. [Accessed: 23-Dec-2020].

[40] "Jenkins." [Online]. Available: https://www.jenkins.io/. [Accessed: 23-Dec-2020].

[41] "Bitbucket Pipelines - Continuous Delivery | Bitbucket." [Online]. Available: https://bitbucket.org/product/features/pipelines. [Accessed: 23-Dec-2020].

[42] "Continuous Integration Tools | Atlassian." [Online]. Available: https://www.atlassian.com/continuous-delivery/continuous-integration/tools. [Accessed: 23-Dec-2020].

[43] "Open Source Continuous Delivery and Release Automation Server | GoCD." [Online]. Available: https://www.gocd.org/. [Accessed: 23-Dec-2020].

[44] "Buddy: The DevOps Automation Platform." [Online]. Available: https://buddy.works/?utm_source=20_best_ci_tools&utm_medium=referral&utm_campaign=guru99. [Accessed: 23-Dec-2020].

[45] "CloudBees CodeShip Features & Pricing." [Online]. Available: https://www.cloudbees.com/codeship/features-pricing#pro-features. [Accessed: 23-Dec-2020].

[46] "Comparison of Most Popular Continuous Integration Tools: Jenkins, TeamCity, Bamboo, Travis CI and more | AltexSoft." [Online]. Available: https://www.altexsoft.com/blog/engineering/comparison-of-most-popular-continuous-integration-tools-jenkins-teamcity-bamboo-travis-ci-and-more/. [Accessed: 23-Dec-2020].

[47] "Bamboo Continuous Integration and Deployment Build Server." [Online].

Available: https://www.atlassian.com/software/bamboo. [Accessed: 23-Dec-2020].

[48] "DevOps Platform Delivered as a Single Application | GitLab." [Online]. Available: https://about.gitlab.com/. [Accessed: 23-Dec-2020].

[49] "Continuous Integration and Delivery - CircleCI." [Online]. Available: https://circleci.com/. [Accessed: 23-Dec-2020].

[50] "AWS CodePipeline | Sürekli Entegrasyon ve Sürekli Teslim." [Online]. Available: https://aws.amazon.com/tr/codepipeline/. [Accessed: 23-Dec-2020].

[51] O. S. Vaidya and S. Kumar, "Analytic hierarchy process: An overview of applications," *Eur. J. Oper. Res.*, vol. 169, no. 1, pp. 1–29, 2006.

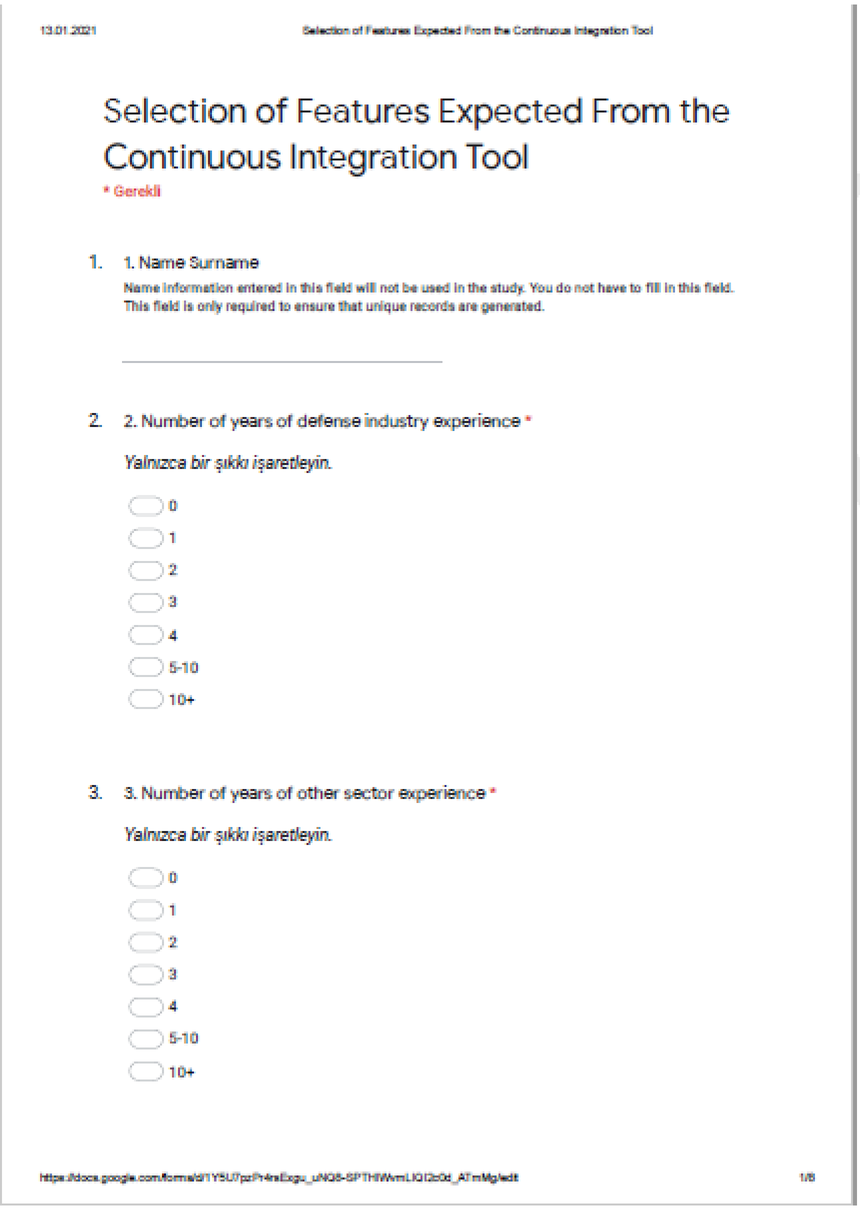
[52] N. Tekin, M. Kosa, M. Yilmaz, P. Clarke, and V. Garousi, "Visualization, Monitoring and Control Techniques for Use in Scrum Software Development: An Analytic Hierarchy Process Approach," *Commun. Comput. Inf. Sci.*, vol. 1251 CCIS, no. August, pp. 45–57, 2020.

[53] T. L. Saaty, "WHAT IS THE ANALYTIC HIERARCHY PROCESS ? Introduction In our everyday life , we must constantly make choices concerning what tasks to do or not to do , when to do them , and whether to do them at all . Many problems such as buying the most cost effective ," 1988.

Part I

Appendices A

Survey of Selection of Features Expected From the Continuous Integration Tool



13.01.2021 Selection of Features Expected From the Continuous Integration Tool

Selection of Features Expected From the Continuous Integration Tool

* Gerekli

1. 1. Name Surname

Name information entered in this field will not be used in the study. You do not have to fill in this field. This field is only required to ensure that unique records are generated.

2. 2. Number of years of defense industry experience *

Yalnızca bir şıkkı işaretleyin.

- 0
- 1
- 2
- 3
- 4
- 5-10
- 10+

3. 3. Number of years of other sector experience *

Yalnızca bir şıkkı işaretleyin.

- 0
- 1
- 2
- 3
- 4
- 5-10
- 10+

https://docs.google.com/forms/d/1Y5U7pzPr4rsExgu_uNQ8-SPTHlWvmLlQl2c0d_ATmMg/edit 1/8

4. 4. Graduated Department *

Yalnızca bir şıkkı işaretleyin.

- ◯ CENG
- ◯ EEE
- ◯ ME
- ◯ Diğer: ____________

5. 5. Post Graduation Department

Yalnızca bir şıkkı işaretleyin.

- ◯ None
- ◯ CENG
- ◯ EE
- ◯ ME
- ◯ Diğer: ____________

6. 6. Working Unit (Department) *

Yalnızca bir şıkkı işaretleyin.

- ◯ Software Simulation
- ◯ Software Test
- ◯ Software Architectural Design
- ◯ Embedded Software

7. 7. Have you ever had continuous integration experience before? *

Yalnızca bir şıkkı işaretleyin.

- ◯ Yes
- ◯ No

Reminding: Continuous Integration

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day.
more info: https://en.wikipedia.org/wiki/Continuous_integration

Reminding: Features provided by continuous integration tools

Features Provided by Continuous Integration	Description
Built-in features	Built features supported language platform (C,C ++, java, C # ,python., Providing build automation tool support (maven ant conan…)
Operating Systems	The operating systems with which the tool runs(Windows, Linux, MacOs, Unix..)
Integration and Software Support & Plugins	Comprehensive plugins & integration support for platforms, UI, administration, source code management, build management (Ide, jira, svn, github, parasoft……)
Container Support.	Having a deployment plug-in or configuration for container editing tools like Kubernetes and Docker makes it easy for a CI tool to connect to the application's target environment.
Docs and Supports	Documents and supporters provided to the user
Easy to use & setup	Ease of installation and use
Use Case	Target project cases, small projects or large projects
Support Continuous Delivery	Continuous Delivery Support Status(yes, no)
License Pricing	Free or price
Hosting Options	Hosting options Cloud or on premise or both
Open Source	the tool is open source or not

https://docs.google.com/forms/d/1Y5U7pzPr4reExgu_uNQ8-SPTHIWvmLIQI2b0d_ATmMg/edit 4/8

13.01.2021 Selection of Features Expected From the Continuous Integration Tool

8. 8. In our institution, choose your expectations from the continuous integration tool features. *

You can choose multiple. If you have suggestions other than options, please specify the other tab. You can also suggest your own customized process metrics. Uygun olanların tümünü işaretleyin.

Her satırda yalnızca bir şıkkı işaretleyin.

	Very necessary	Necessary	Neutral	Not so necessary	Not at all necessary
Windows Platform Support (The tool can run on Windows operating system)	⬭	⬭	⬭	⬭	⬭
Linux Platform Support (The tool can run on Linux operating system)	⬭	⬭	⬭	⬭	⬭
Unix Platform Support (The tool can run on Unix operating system)	⬭	⬭	⬭	⬭	⬭
MacOS platform Support (The tool can run on MacOS operating system)	⬭	⬭	⬭	⬭	⬭
Build Support of C/C++ (The tool provides build in C/C ++ language)	⬭	⬭	⬭	⬭	⬭
Build Support of Java (The tool provides build in java language)	⬭	⬭	⬭	⬭	⬭
Build Support of C# (The tool provides build in C# language)	⬭	⬭	⬭	⬭	⬭
Build Support of Python (The tool provides build in python language)	⬭	⬭	⬭	⬭	⬭
On premise support (Offline) (For intranet, the tool keeps projects on server and allows offline use)	⬭	⬭	⬭	⬭	⬭
Maven Support of build automation Tool (The tool allows the use of the	⬭	⬭	⬭	⬭	⬭

https://docs.google.com/forms/d/1Y5U7pzPr4raExgu_uNQ8-SPTHWvmLIQI2c0d_ATmMg/edit 5/8

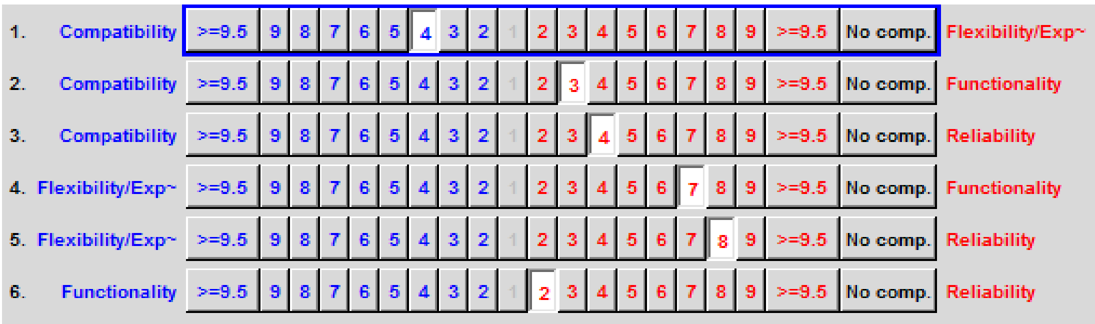
Maven build automation tool)					
Conan Support of build automation Tool (The tool allows the use of the Conan build automation tool)					
Ant Support of build automation Tool (The tool allows the use of the Ant build automation tool)					
Ease of use and Setup (easy installation and use of the tool)					
Easy configurability (Tool can be easily configured)					
Support of Continuous Delivery (The tools provide CD)					
E-mail Notification (The tool notify the user by mail)					
Open Source (The tool is open source so suitable for development)					
Support of Tests(mstest, jtest...) plugins (Tool supports various test frameworks)					
Support of Test Result, test analysis and report (The tool shows the test results, can analyze and report them.)					
Support a large plug-in resource (The tool should have a large source of plugins so it can meet the extra needs))					
Just use for small projects (The tool is suitable for					

Part II

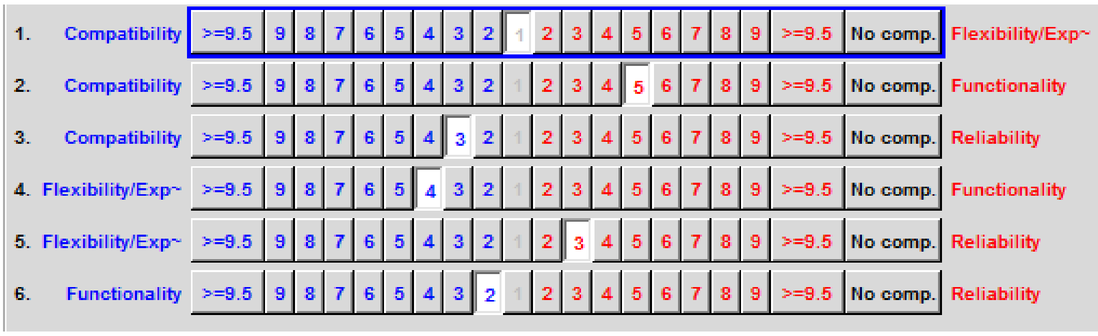
Appendices A
Appendices

AHP Implementations Results

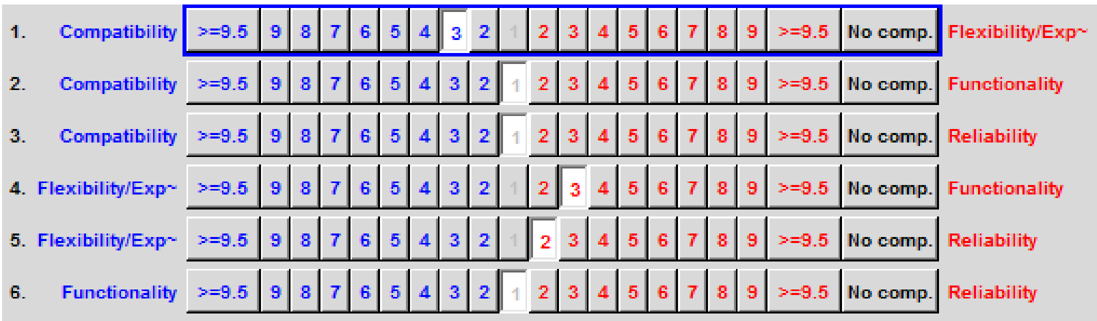
Pairwise Comparison of criteria result –Expert1



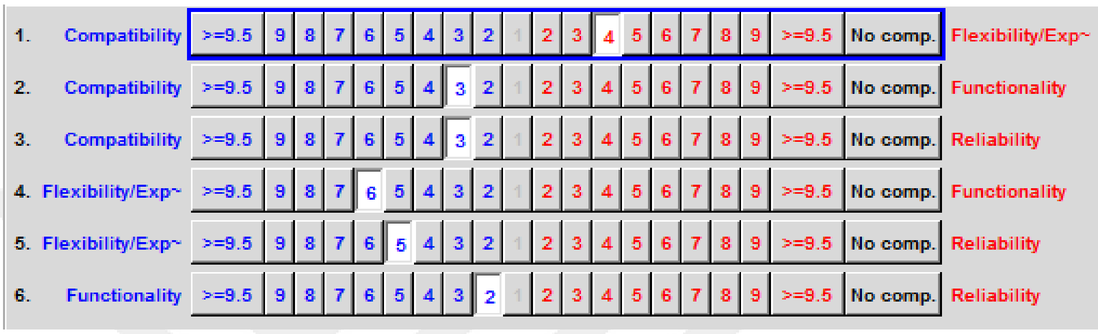
Pairwise Comparison of criteria result –Expert2



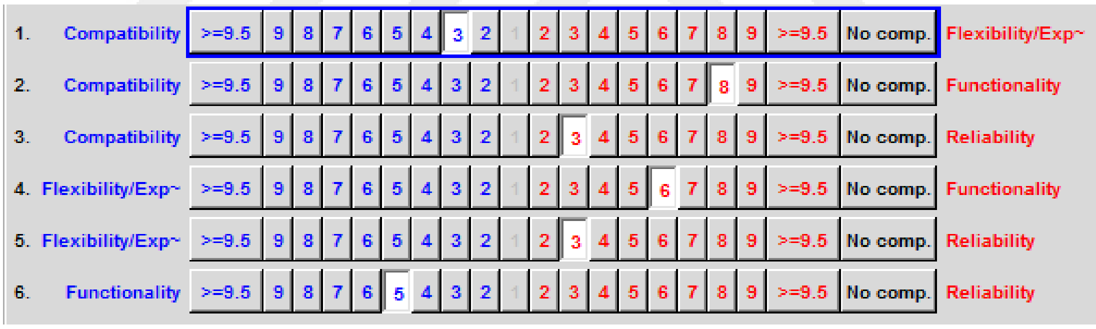
Pairwise Comparison of criteria result –Expert3



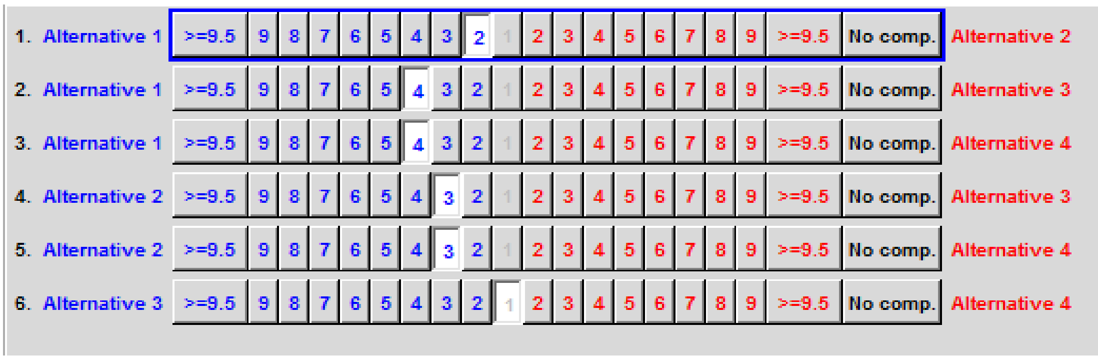
Pairwise Comparison of criteria result –Expert4



Pairwise Comparison of criteria result –Expert5



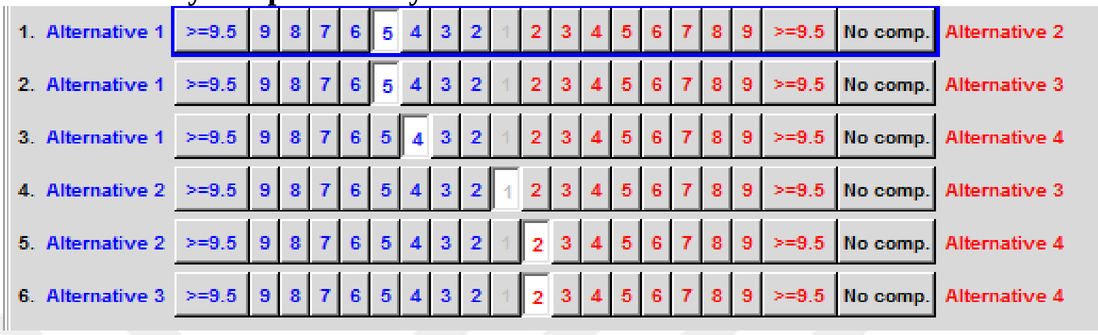
CI Tool alternatives Evaluation- Expert1
For Compatibility criteria:



Normalization values alternatives for criteria Compatibility:

Alternative 1	0.48519
Alternative 2	0.29682
Alternative 3	0.10899
Alternative 4	0.10899

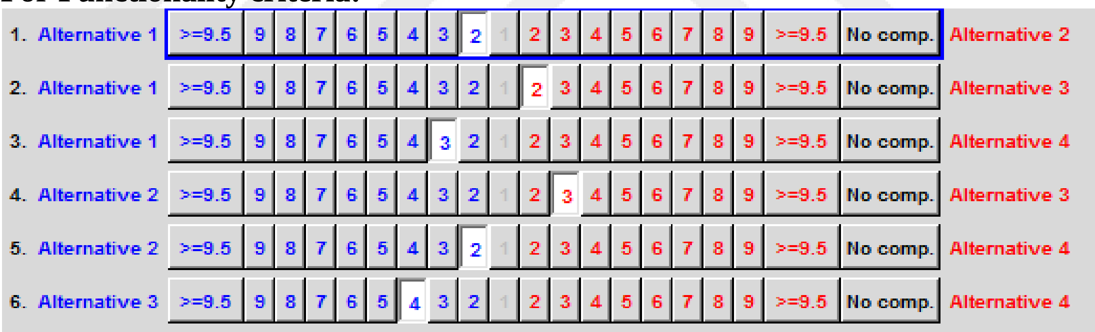
For FlexibiIity&Expandability criteria:



Normalization values alternatives for criteria FlexibiIity&Expandability:

Alternative 1	0.59927
Alternative 2	0.10576
Alternative 3	0.10576
Alternative 4	0.18921

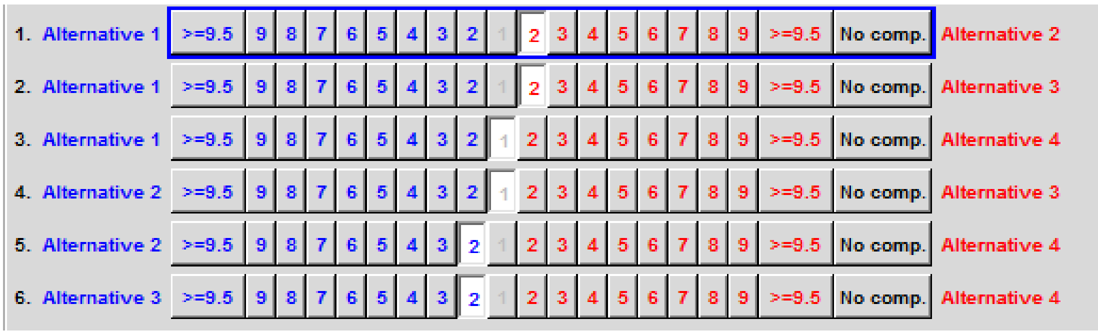
For Functionality criteria:



Normalization values alternatives for criteria Functionality:

Alternative 1	0.27718
Alternative 2	0.16009
Alternative 3	0.4673
Alternative 4	0.09543

For Reliability criteria:

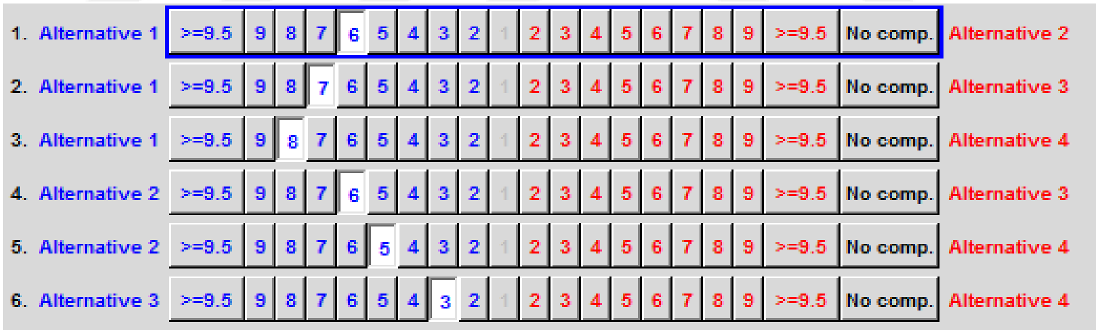


Normalization values alternatives for criteria Reliability:

Alternative 1	0.16667
Alternative 2	0.33333
Alternative 3	0.33333
Alternative 4	0.16667

CI Tool alternatives Evaluation- Expert2

For Compatibility criteria:



Normalization values alternatives for criteria Compatibility:

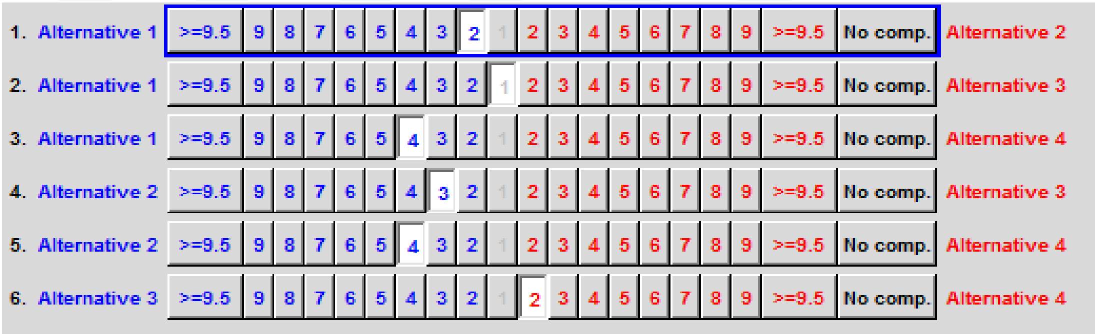
Alternative 1	0.65196
Alternative 2	0.22768
Alternative 3	0.07624
Alternative 4	0.04412

For Flexibility&Expandability criteria:

Normalization values alternatives for criteria FlexibiIity&Expandability:

Alternative 1	0.60201
Alternative 2	0.22109
Alternative 3	0.09091
Alternative 4	0.08599

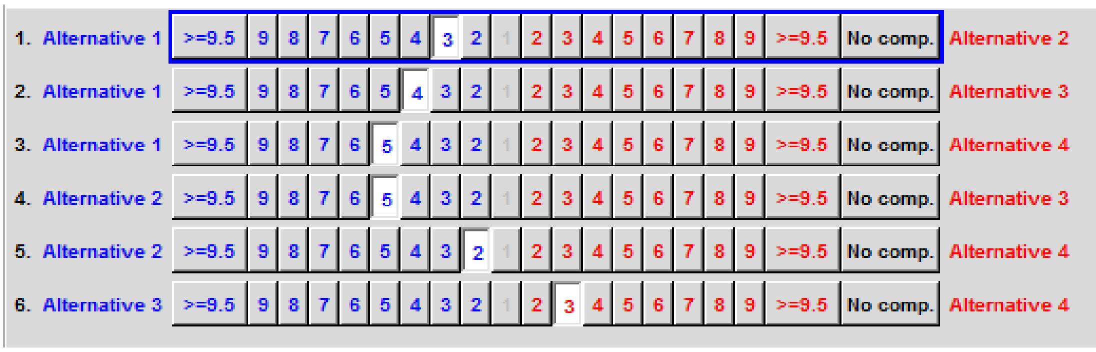
For Functionality criteria:



Normalization values alternatives for criteria Functionality:

Alternative 1	0.37803
Alternative 2	0.33199
Alternative 3	0.15466
Alternative 4	0.13533

For Reliability criteria:



Normalization values alternatives for criteria Reliability:

Alternative 1	0.53757
Alternative 2	0.25213
Alternative 3	0.07154
Alternative 4	0.13876

CI Tool alternatives Evaluation- Expert3

For Compatibility criteria:

1.	Alternative 1	>=9.5	9	8	7	6	5	4	**3**	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 2
2.	Alternative 1	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
3.	Alternative 1	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
4.	Alternative 2	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
5.	Alternative 2	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
6.	Alternative 3	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4

Normalization values alternatives for criteria Compatibility:

Alternative 1	0.33101
Alternative 2	0.18759
Alternative 3	0.2407
Alternative 4	0.2407

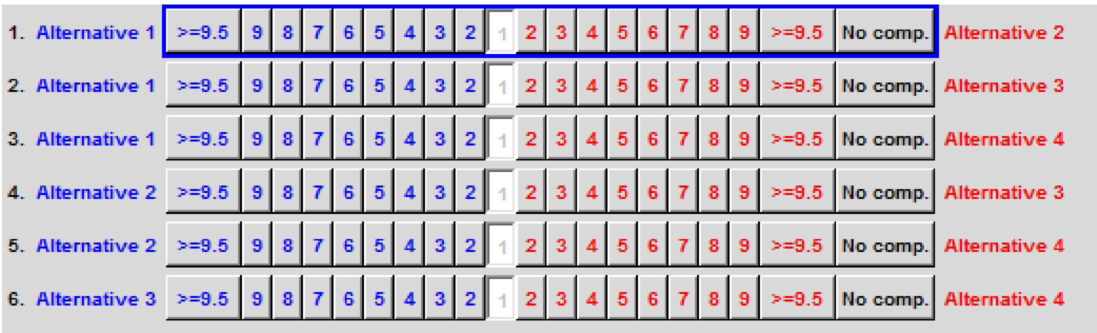
For FlexibiIity&Expandability criteria:

1.	Alternative 1	>=9.5	9	8	7	6	5	**4**	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 2
2.	Alternative 1	>=9.5	9	8	7	6	5	4	**3**	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
3.	Alternative 1	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
4.	Alternative 2	>=9.5	9	8	7	6	5	4	3	2	1	2	**3**	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
5.	Alternative 2	>=9.5	9	8	7	6	5	4	3	2	1	2	**3**	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
6.	Alternative 3	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4

Normalization values alternatives for criteria FlexibiIity&Expandability:

Alternative 1	0.4111
Alternative 2	0.0865
Alternative 3	0.21797
Alternative 4	0.28442

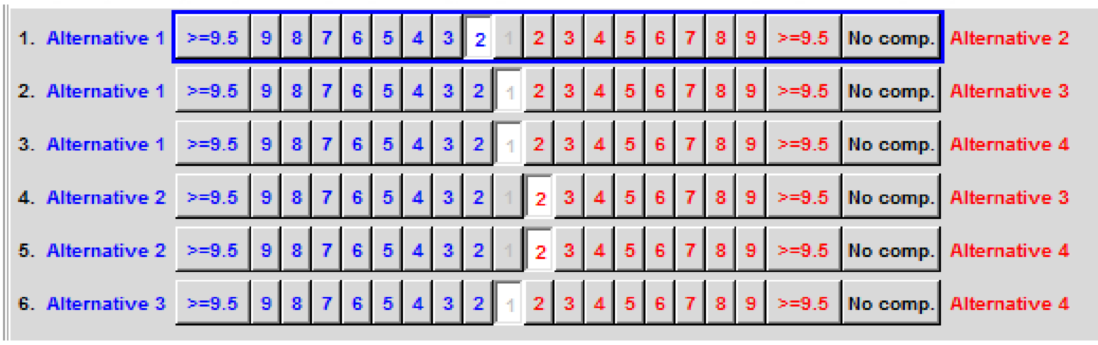
For Functionality criteria:



Normalization values alternatives for criteria Functionality:

Alternative 1	0.25
Alternative 2	0.25
Alternative 3	0.25
Alternative 4	0.25

For Reliability criteria:

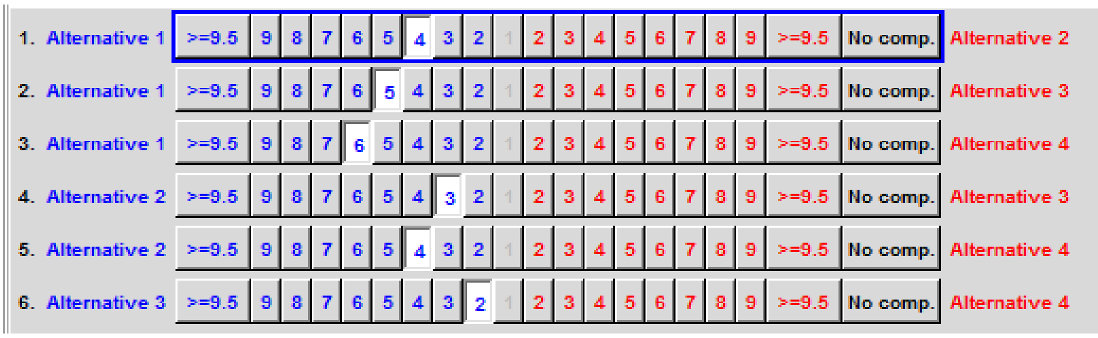


Normalization values alternatives for criteria Reliability:

Alternative 1	0.28571
Alternative 2	0.14286
Alternative 3	0.28571
Alternative 4	0.28571

CI Tool alternatives Evaluation- Expert4

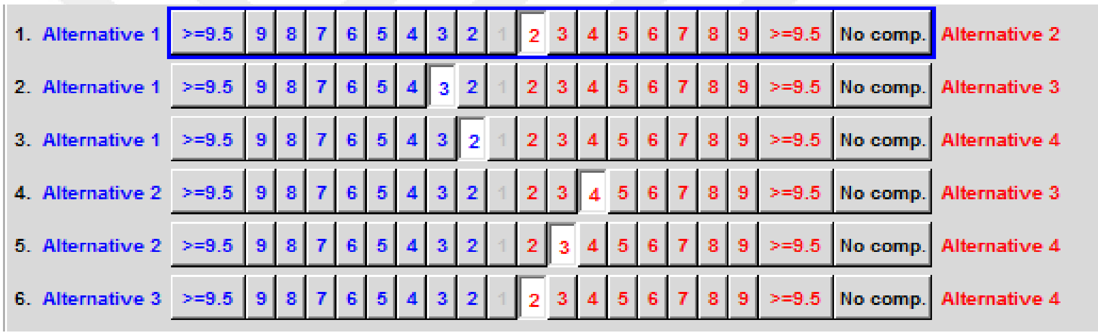
For Compatibility criteria:



Normalization values alternatives for criteria Compatibility:

Alternative 1	0.59403
Alternative 2	0.23358
Alternative 3	0.10541
Alternative 4	0.06698

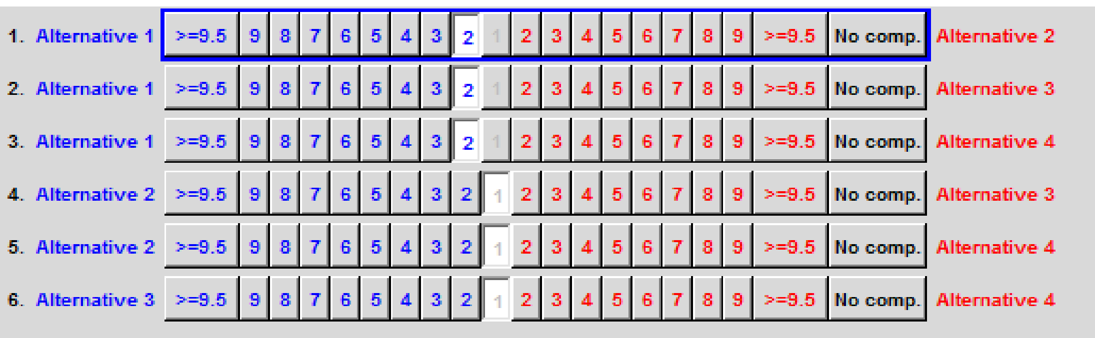
For FlexibiIity&Expandability criteria:



Normalization values alternatives for criteria FlexibiIity&Expandability:

Alternative 1	0.31328
Alternative 2	0.18283
Alternative 3	0.22943
Alternative 4	0.27446

For Functionality criteria:



Normalization values alternatives for criteria Functionality:

Alternative 1	0.4
Alternative 2	0.2
Alternative 3	0.2
Alternative 4	0.2

For Reliability criteria:

1. Alternative 1	>=9.5	9	8	7	6	5	4	3	2	1	2	**3**	4	5	6	7	8	9	>=9.5	No comp.	Alternative 2
2. Alternative 1	>=9.5	9	8	7	6	5	4	3	2	1	**2**	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
3. Alternative 1	>=9.5	9	8	7	6	5	4	3	**2**	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
4. Alternative 2	>=9.5	9	8	7	6	5	4	3	**2**	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
5. Alternative 2	>=9.5	9	8	7	6	5	**4**	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
6. Alternative 3	>=9.5	9	8	7	6	5	4	**3**	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4

Normalization values alternatives for criteria Reliability:

Alternative 1	0.16009
Alternative 2	0.4673
Alternative 3	0.27718
Alternative 4	0.09543

CI Tool alternatives Evaluation- Expert5

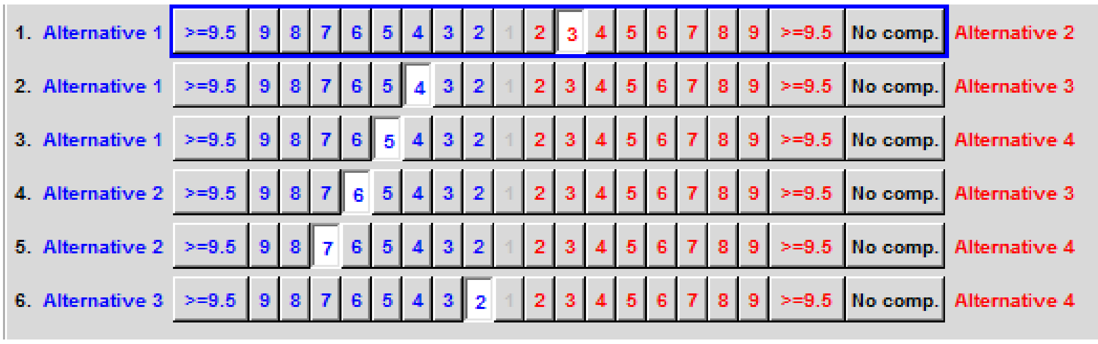
For Compatibility criteria:

1. Alternative 1	>=9.5	9	8	7	6	5	4	3	2	**1**	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 2
2. Alternative 1	>=9.5	9	8	7	6	5	**4**	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
3. Alternative 1	>=9.5	9	8	7	6	5	**4**	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
4. Alternative 2	>=9.5	9	8	7	**6**	5	4	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 3
5. Alternative 2	>=9.5	9	8	7	**6**	5	4	3	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4
6. Alternative 3	>=9.5	9	8	7	6	5	4	**3**	2	1	2	3	4	5	6	7	8	9	>=9.5	No comp.	Alternative 4

Normalization values alternatives for criteria Compatibility:

Alternative 1	0.36788
Alternative 2	0.45346
Alternative 3	0.11393
Alternative 4	0.06473

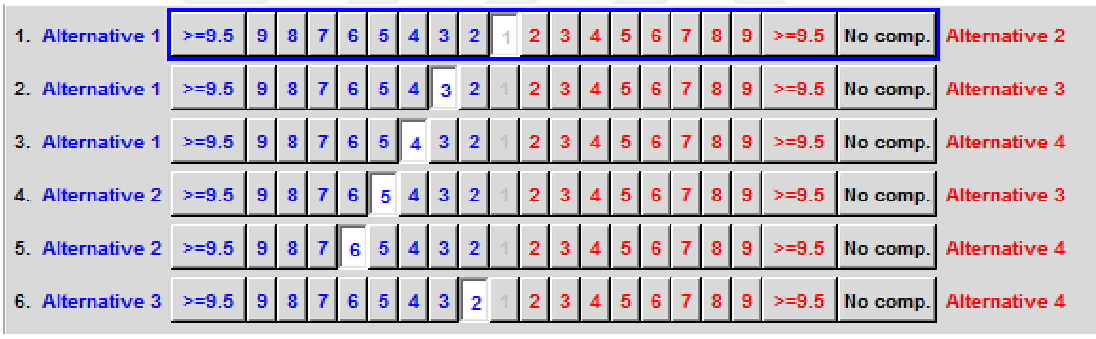
For FlexibiIity&Expandability criteria:



Normalization values alternatives for criteria FlexibiIity&Expandability:

Alternative 1	0.27459
Alternative 2	0.57535
Alternative 3	0.09113
Alternative 4	0.05894

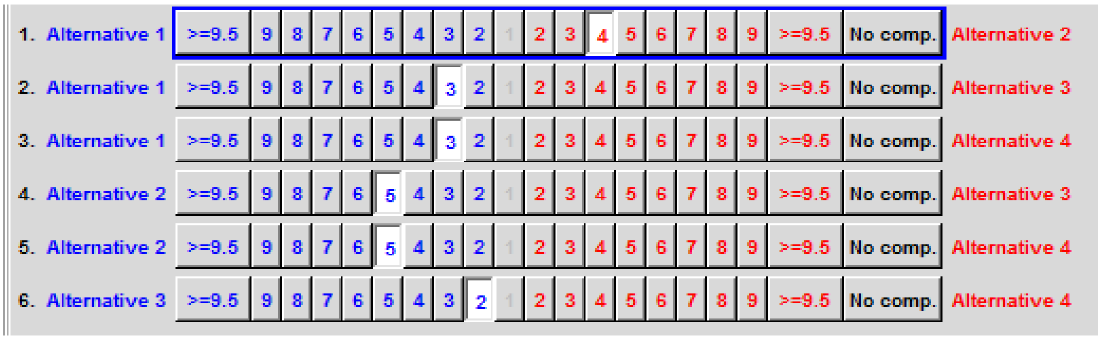
For Functionality criteria:



Normalization values alternatives for criteria Functionality:

Alternative 1	0.35834
Alternative 2	0.45195
Alternative 3	0.1166
Alternative 4	0.0731

For Reliability criteria:



Normalization values alternatives for criteria Reliability:

Alternative 1	0.22583
Alternative 2	0.58521
Alternative 3	0.11055
Alternative 4	0.0784

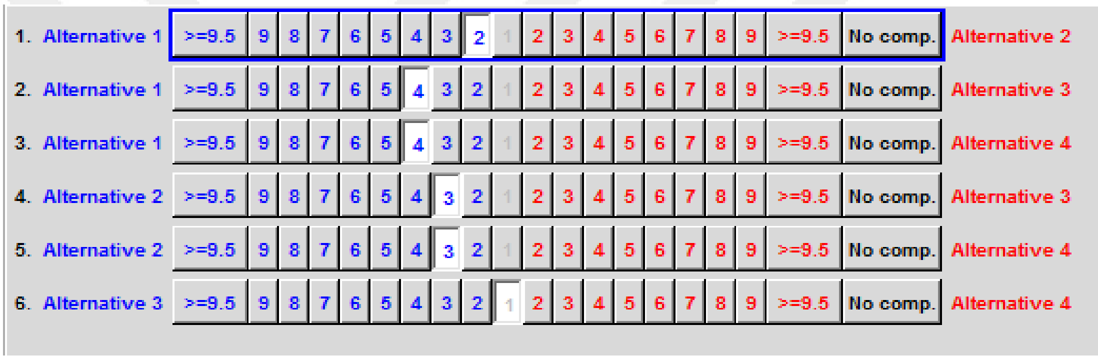
Manuel Calculation of AHP process

	Alt1	Alt2	Alt3	Alt 4
pairwise	c1	c2	c3	c4
c1	1	2	4	4
c2	1\2	1	3	3
c3	1\4	1\3	1	1
c4	1\4	1\3	1\1	1

	Alt1	Alt2	Alt3	Alt 4
pairwise	c1	c2	c3	c4
c1	1	2	4	4
c2	0,5	1	3	3
c3	0,25	0,333333	1	1
c4	0,25	0,333333	1	1
total	2	3,666667	9	9

	Alt1	Alt2	Alt3	Alt 4
pairwise	c1	c2	c3	c4
c1	1\2	2\3,666667	4\9	4\9
c2	0,5\2	1\3,666667	3\9	3\9
c3	0,25\2	0,333333\3,666667	1\9	1\9
c4	0,25\2	0,333333\3,666667	1\9	1\9

	Alt1	Alt2	Alt3	Alt 4	
pairwise	c1	c2	c3	c4	total
c1	0,5	0,545455	0,444444	0,444444	0,483585859
c2	0,25	0,272727	0,333333	0,333333	0,297348485
c3	0,125	0,090909	0,111111	0,111111	0,109532828
c4	0,125	0,090909	0,111111	0,111111	0,109532828



Alternative 1	0.48519
Alternative 2	0.29682
Alternative 3	0.10899
Alternative 4	0.10899

Alternative1 :JENKINS

Jenkins is the number one open-source project for automating your projects. With thousands of plugins to choose from, Jenkins can help teams to automate any task that would otherwise put a time-consuming strain on your software team. Common uses include building projects, running tests, bug detection, code analysis, and project deployment. This software helps developers to quickly find and solve defects in their code base & automate testing of their builds.

Jenkins key features:

- Easy installation and upgrade to various operating systems
- Simple and easy to use interface
- Extensible with a huge community-based plugin resource
- Easy configuration of the environment in the user interface
- Supports distributed master-slave architecture builds
- Build schedules based on phrases

- Supports execution of Windows shells and commands in pre-build steps
- Supports notification of build status

License: Free. Jenkins is an open-source tool with an active community.

Alternative 2: Teamcity

TeamCity from Jetbrains is an intelligent CI server solution for software environments of all sizes. With an ample amount of features integrated specifically for developers, you will not be disappointed by the level of performance that TeamCity adds to your team. Built to support modern software stacks and platforms, you can get started within minutes using pre-built installers.

TeamCity key features:

- Provides several ways for the subproject to reuse parent project settings and configurations
- The parallel runs works on various environments simultaneously
- Allows to build history, view test history reports, pin, tag and add favourites
- Easy to customize, interact and server extension
- Keeps the CI Server stable and functional
- Flexible user management, assignment of user roles, grouping of users, different user authentication methods and a log with all user actions to ensure transparency of all server activities

License: TeamCity is a commercial tool with both free and proprietary licenses.

Alternative 3:Bamboo

Bamboo is a CI server being used by software teams across the globe to automate the process of release management for applications and general software, allowing teams to establish a streamlined pipeline of build delivery. Bamboo gives developers a chance to automate their build and test processes, in turn freeing up time that can be spent improving the product itself. Mobile developers can deploy their apps back to the Apple Store or Google Play automatically.

Bamboo key features:

- Provides support for up to 100 remote agents
- Run test batches in parallel and get quick feedback
- Creates images and pushes to a record
- Per-environment permissions which allow developers and testers to deploy on demand in their environments while the output remains locked
- Detects new branches in Git, Mercurial, SVN Repos and automatically applies the main line CI scheme to them
- Triggers build based on modifications found in the repository. Pushes Bitbucket notifications, a set schedule, completing another build or any combination thereof.

License: Bamboo price ranges are based on agents rather than users, or “build slaves.” The more agents it can run at the same time, the more processes it can run-either in the same build or different builds.

Alternative 4: Buddy

Besides the beautiful and rich user interface that Buddy web platform is rocking, you get a high-quality service for automating your development, without the complexity of using custom tools to do so. Buddy’s pride is simplicity, and it shines through their automated pipeline feature which helps developers to test, build and ship their software to production quicker than ever before.

Buddy key features:

- Fast to customize Docker based images as an environment for testing
- Detection of intelligent improvement, state-of-the-art caching, parallelism and all-round optimisation
- Develop and test environments, build, customize and reuse
- The scopes are simple and encrypted, fixed and settable: workspace, mission, pipeline, acts
- Services available with Elastic, MariaDB, Memcached, Mongo, PostgreSQL, RabbitMQ, Redis, Chrome Selenium, and Firefox
- Monitor the progress and the logs in real time, unlimited history
- Managing workflows with models for cloning, exporting and import pipelines
- Support for and integration of first class Git

License: Buddy is a commercial free tool.

	Jenkins	TeamCity	Bamboo	Buddy
Open-source	Yes	No	No	Yes
Free version	Yes	Yes	Yes	Yes
Supported operating systems	Windows, Linux, macOS, any Unix-like OS	Windows, Linux, macOS, Solaris, FreeBSD	Windows, Linux, macOS, Solaris	Linux, Windows, macOS
Cloud hosting	Yes	No	Yes	Yes
On-premise hosting	Yes	Yes	Yes	Yes
Ease of setup	Very easy (has pre-built packages)	Very easy (has pre-built packages)	Easy	Very easy (15-minute setup with a range of predefined actions)
Main feature	Hundreds of plugins	Gated commits	Very user-friendly	Extremely easy to create, adjust, and manage