CHAPTER 3

INTRODUCTION TO AUTOMATON

# 3.1 Introducing the Adaptive (Automated) Life Cycle Management (ALM)

The present applications confront the test of doing required errands and perform inside a regularly contracting calendar and with negligible assets. Thus Project partners endeavour to accomplish more with less, on the grounds that associations need to test programming as fast and altogether as would be prudent. Subsequently associations are swinging to computerized testing for Applications, Systems and Other Hardware assets.

This is reality that many (a large portion of the) tests can't be executed physically constantly, for instance mimicking 1, 0000 virtual clients for volume testing, computerized testing is presented for such tasks. Yet, partners may not realize what's associated with acquainting a computerized apparatus with a product venture, and furthermore new to the broadness of utilization that robotized devices have.

In any case, by characterizing orderly methodology inside the ALM, test exercises can be sorted out and executed so as to amplify test inclusion inside the cut-off points of testing assets. Organized test approach includes a multi-arrange process, supporting the itemized and interconnected exercises that are required to present and use a robotized test instruments and Techniques:

1. Develop test design and test cases.
2. Develop and execute test cases.
3. Develop, manage and arrange test data and the test environment.
4. Document, track, and obtain closure on issue/trouble reports.

Automated testing speaks to a worldview change in the product business as well as in assembling Industry. This change doesn't just include the use of test computerization apparatuses. Or maybe, it have the whole test lifecycle and the framework advancement lifecycle. The ALM usage happens in parallel with the framework (programming or any item) improvement lifecycle. Programming experts (engineers) to make a fruitful jump to robotized versatile testing, they should grasp organized ways to deal with testing. The ALM is progressive as it declares another organized, assembling square way to deal with the whole test lifecycle, which empowers programming experts and partners to approach programming testing and Automation in an orderly and repeatable form.

The growth of adaptive capability has stemmed in large part from the growing popularity of the iterative and incremental development lifecycle, a software development methodology that focuses on minimizing the development schedule while providing frequent, incremental software builds. The objective of incremental and iterative development is to engage the user early throughout the design and development of each build in order to refine the software, and ensuring that it more closely reflects the needs and preferences of the user and thus addressing the required aspects of development in early builds.

In the environment of continuous changes and updates to the software through each software build, software testing itself takes on an iterative nature. Next build is joined by a significant number of extra tests just as adjust to existing test contents, or in programming modules. Given the nonstop changes and increases to programming applications, particularly Web applications, versatile programming testing turns into a vital control component to guarantee precision and soundness of the product through each form of use.

The ALM, summoned to help mechanization endeavours including computerized test devices, joins a multi-organize process. This backings the point by point and interrelated exercises that are required to choose about Tools and Methods required for venture.

The methodology includes the process of how to utilize an automated test tool, covers test development and test design, and addresses test execution and management. It also supports the development and management of test data and the test environment, and addresses test documentation to include problem reports.

It represents a structured approach to execute automated testing. It is required to push the team far from these common test (White Box, Black Box) program mistakes:

1. Starts the utilization of automated test apparatus without a testing procedure set up, bringing about a specially appointed, non-repeatable, non-quantifiable test program leaves from versatile methodology
2. Starts Implementation of a test plan without following any structure principles, bringing about the formation of test contents that are not repeatable and in this manner not reusable for steady programming fabricates.
3. Doing Attempt to computerize 100% of testing necessities, when devices or in-house– created mechanized test bridles don't bolster automation of all tests required.
4. Using the wrong device or creating explained in-house test tackle.
5. Including test instrument execution past the point of no return in the Application-Development Lifecycle, consequently not permitting adequate time for device setup and test apparatus presentation process (expectation to absorb information) which ought to be of Adaptive nature.
6. Initiating Automation Test engineer involvement too late in the application-development lifecycle, resulting in poor understanding of the application and system design, which results in incomplete automation testing.

The Adaptive (Automated) Lifecycle Methodology (ALM) comprises following components:

1. Take Decision to Automate in Quality Assurance, Quality Engineering and Build & Release.
2. Getting requited Tools Acquisition for Analysis.
3. Doing Automated Kick-off Process.
4. Working with Planning, Design, and Development.
5. Defining, Execution with Management of Process.
6. Finalize Automation Revisit and Assessment of process.

Next sections details about process, and subordinate processes contained within each primary process

# 3.2 Decision to Automation

The choice to mechanization speaks to the principal period of the ALM. This stage covers the whole procedure that goes into the computerization choice. Amid this stage, it's vital for the group to oversee computerization desires and to layout the potential advantages of automation when executed accurately. An apparatus proposition should be delineated, which will be useful in securing the board bolster.

## 3.2.1 Overcoming Undesired Expectations for Automated Testing

While it has been demonstrated that automation is profitable and can deliver an effective rate of profitability, there isn't generally a prompt restitution on speculation. It's essential to address a portion of the misguided judgments that hold on in the business and to deal with the automation perfect world. There is a misinterpretations.

Computerization needs a noteworthy transient speculation of time and vitality to accomplish a long haul rate of return (ROI) of quicker and less expensive relapse.

## 3.2.2 Test Plan Generation Automation

Right now, there is no industrially accessible apparatus that can naturally make a thorough arrangement while additionally supporting structure and execution.

All through a product vocation, the designer can hope to observe apparatus exhibitions and survey a plenteous measure of hardware writing. Regularly the architect will be requested to remain before at least one senior directors to give an apparatus usefulness review. For this situation, the gathering of people may speak to people with simply enough specialized information to make them eager about automation, while unconscious of the multifaceted nature required with a mechanization exertion. In particular, the supervisors may have acquired used data about mechanized instruments, and may have achieved the wrong translation of the real capacity of robotized devices.

The executives may expect that the instrument being proposed consequently builds up the arrangement, structures and makes the systems, executes every one of the methods, and examines the outcomes naturally. In the meantime, illuminating the gathering that computerized instruments ought to be seen as improvements to manual process, and that robotized devices won't build up the arrangement, structure and make the methodology, or execute the systems.

In the blink of an eye into the introduction and after a few administration questions, it turns out to be exceptionally clear exactly the amount of a separation exists between the truth of the apparatus abilities and the impression of the people in the group of onlookers. The mechanized device appears to carry with it a lot of unrealistic reasoning that is not firmly lined up with the real world.

A robotized apparatus won't trade the human factor vital for testing an item. The proficiencies of specialists and other QA specialists will at present be expected to keep the test hardware running. A computerization device can be seen as an extra piece of the hardware that underpins the arrival of a decent item.

## 3.2.3 One Test Automation Tool Fits All Scenarios

A solitary apparatus can't be utilized to help all working framework situations.

For the most part, a solitary device won't satisfy every one of the prerequisites for an association/group. Consider the experience of one architect experiencing such a circumstance. The specialist is asked by a supervisor to discover a device that could be utilized to mechanize the all the division's applications. The office utilizing different advances including centralized server PCs and Sun workstations; working frameworks, for example, programming dialects, for example, C,C++, Java, Python other customer/server innovations; and Web advances, for example, DHTML, XML, ASP, etc.

In the wake of directing an apparatus assessment, the specialist discovered that the device of decision was not good with the Visual C++ outsider additional items (for this situation, Stingray lattices). Other device to be acquired that was perfect with this explicit application.

## 3.2.4 Immediate Change in Schedule

An automated device will dependably not limit the testing schedule. Another mechanization misinterpretation is the desire that the utilization of a computerized apparatus on another venture will promptly limit the timetable. The timetable won't encounter the foreseen decline at first, and a stipend for calendar increment is required when at first presenting a computerized apparatus. This is because of the way that when revealing a robotized device, the present procedure must be enlarged or a completely new process must be produced and executed.

The whole group and perhaps the improvement group, needs to get comfortable with this new computerized process, (for example, ALM) and requirements to tail it. When a programmed procedure has been set up and successfully actualized, the venture can hope to encounter gains in profitability and turnaround time that positively affect calendar and cost.

## 3.2.5 Advantage of Automated Test Execution, Quality Engineering, And Continues Integration/Delivery.

The past discourse brings up and illuminates a portion of the false mechanized desires that exist. The specialist will likewise should have the capacity to expound on the genuine advantages of automation, when mechanized testing is executed effectively and a procedure is pursued. The architect must assess whether potential advantages fit required enhancement criteria and whether the quest for mechanization on the undertaking is as yet a legitimate fit, given the authoritative needs. There are three critical computerization benefits:

• Producing a dependable framework.

• Improving the nature of the exertion.

• Reducing exertion and limiting calendar.

Many degree of profitability contextual analyses have been finished as to the execution of mechanization.

## 3.2.6 Getting Management Support

At whatever point any association attempts to receive another innovation, they experience a huge exertion while deciding how to apply it to their requirements. With some preparation, associations grapple with tedious false begins previously they wind up competent with the new innovation. For the group keen on executing robotized devices, the test is the means by which to best present the case for another automation innovation and its usage to the supervisory crew.

Designers need to impact the executives' desires for the utilization of mechanization on ventures. Specialists can oversee desires for others in the association by sending accommodating data to the administration staff. Raising device issues amid system and arranging gatherings can likewise help grow better comprehension of hardware capacities for everybody required on an undertaking or inside the association. An architect can create preparing material regarding the matter of mechanization and can propose to the board that a workshop be booked to lead the preparation.

The initial phase in advancing toward a choice to computerization on a venture necessitates that the group ought to have the capacity to show to the board comprehension of the suitable use of automation for the explicit need within reach. For instance, the group needs to check at an opportune time whether the board is cost-disinclined and would be reluctant to acknowledge the assessed expense of computerized instruments for a specific exertion. Provided that this is true, official need to persuade the executives about the potential degree of profitability by directing cost/advantage investigation.

In the event that administration will put resources into a mechanized device, however can't or reluctant to staff a group with people having the best possible programming ability level or to accommodate satisfactory apparatus preparing, the group needs to call attention to the dangers included or potentially may need to rethink a proposal to automation.

The board should be made mindful of the extra cost included while presenting another device, for the device buy, as well as for introductory calendar/cost increment, extra preparing expenses, and for improving a current procedure or executing another procedure.

Mechanization speaks to adaptable innovation, which gives a few different ways to achieve a target. Utilization of this innovation requires better approaches for considering, which just enhances the issue of hardware usage. The issues that associations confront while embracing computerized frameworks incorporate those delineated underneath:

* Finding/hiring tool experts.
* Experimenting/Using the correct tool for the task at hand.
* Developing and implementing an automated process, which includes developing automated design and development standards.
* Analyzing applications to determine those that are best suited for automation.
* Analyzing the requirements to determine suitability for automation.
* Training the team on the automated process, automated design, development, and execution.
* Initial increase in schedule and cost.

# 3.3 Automation Tool Acquisition

Tool acquisition represents the second phase of the ALM. This phase guides the engineer through the entire tool evaluation and selection process, starting with confirmation of management support. Since a tool should support most of the organizations' testing requirements, whenever feasible the engineer will need to review the system's engineering environment and other organizational needs and come up with a list of tool evaluation criteria enabling the reader to make an informed decision with regard to the types of tests to be performed on a particular project. The engineer then needs to define an evaluation domain to pilot the tool. Finally, after all those steps have been completed, engineer can make vendor contact to bring in the selected tool(s). Then tool is evaluated, with sample criteria provided.

# 3.4 Automation (Adaptive) Introduction Process

The process of introducing automation to a new project team constitutes the third phase of the ALM. This phase outlines the steps necessary to successfully introduce automation to a new project, which are summarized in the following sections.

## 3.4.1 Analysis of Automation Process

Process investigation guarantees that a general procedure and technique are set up and are altered, if important, to enable automation to be presented in a fruitful manner. The architects characterize and gather test process measurements so as to take into consideration process enhancement. Here test objectives/targets and methodologies should be characterized and test process should be recorded and imparted to the group. In this stage, the sorts of process pertinent for the specialized condition will be characterized, that can be upheld via computerized instruments.

Amid the procedure investigation, systems are characterized. Plans for client association are surveyed, and group work force aptitudes are broke down against prerequisites and arranged exercises. Early group investment is stressed, supporting refinement of necessity particulars into terms that can be satisfactorily examined while likewise supporting group comprehension of utilization prerequisites and plan.

## 3.4.2 Considering Automation Tool

The instrument thought process incorporates steps that research whether consolidation of mechanized devices that have been brought into the organization without an explicit venture at the top of the priority list presently would be helpful to an explicit undertaking, given the task necessities, accessible condition, work force assets, client condition, stage, and item highlights of the application under execution. Calendar is assessed to guarantee adequate time for instrument setup and improvement of necessities pecking order; potential devices and utilities are mapped to prerequisites, device similarity with the application and condition is checked, and workaround arrangements are found for contradiction issues.

# 3.5 Automation Planning with Design, And Development

Arranging, structure, and advancement is the fourth period of the ALM. These subjects are outlined in the accompanying areas.

## 3.5.1 Automation System Planning

The arranging stage speaks to the need to audit long– lead-time test arranging exercises. Amid this stage, the test group recognizes technique creation models and rules; equipment, programming, and system required to help condition; test information necessities; a starter plan; execution measure prerequisites; a strategy to control arrangement and condition; just as deformity following procedure(s) and related following tool(s).

The arrangement contains the aftereffects of every primer period of the organized strategy (ALM). The arrangement will characterize jobs and duties, venture plan, test arranging and structure exercises, condition readiness, dangers and possibilities, and worthy dimension of careful quality (acknowledgment criteria). Plan supplements may incorporate systems, naming traditions, technique arrange benchmarks, and a method detectability lattice.

Nature setup is a piece of arranging. It speaks to the need to plan, track, and oversee condition setup exercises, where material obtainments may have long lead times. The group needs to timetable and track condition setup exercises; introduce condition equipment, programming, and system assets; coordinate and introduce condition assets; get/refine databases; and create condition setup contents and proving ground contents.

## 3.5.2 Automation System Design

The plan part delivers the need to characterize the quantity of criteria to be played out, the manners in which that mechanization will be drawn closer (ways, capacities), and the conditions that should be worked out. Structure guidelines should be characterized and pursued.

A viable program, fusing the computerization of programming testing, includes a smaller than expected advancement lifecycle of its own, total with system and objective arranging, test prerequisite definition, investigation, plan, and coding. Like programming application advancement, prerequisites must be indicated before configuration is built. Necessities should be plainly characterized and reported, so all venture faculty will comprehend the premise of the exertion. Necessities are characterized inside prerequisite explanations as a result of necessity investigation.

After necessities have been inferred utilizing the depicted systems, technique configuration can start. Methodology configuration comprises of the meaning of consistent gatherings of techniques and a naming tradition for the suite of strategies. With a method definition set up, every methodology is then distinguished as either a computerized or a manual test. Amid the arranging stage, the group gets a comprehension of the quantity of strategies being utilized and a gauge for the quantity of methods that will be required. The group additionally will have a gauge of the quantity of methodology that should be performed physically, just as with a mechanized device.

Much like a product improvement exertion, the program must be mapped out and deliberately intended to guarantee that exercises performed speak to the most proficient and compelling procedure for the framework under execution. Program assets are constrained, yet methods for executing the framework are interminable. A plan is produced to depict the exertion, so as to give undertaking and test work force a psychological structure on the limit and extent of the program.

Following examination, the group builds up the program configuration models. The first of these plan models, the program display, comprises of a graphical outline that delineates the extent of the program. This model regularly delineates the procedures required to help the dynamic exertion and furthermore diagram static methodologies.

Having characterized a program demonstrate, the group builds an engineering, which delineates the structure of the program and characterizes the manner in which that strategies will be sorted out in help of the test exertion.

The following stage in the system configuration process (see Table 3.1) is to distinguish those strategies that emerge as being increasingly modern, and accordingly are required to be characterized further as a feature of itemized plan. These systems are hailed and an itemized plan record is set up in help of the more advanced techniques.

Following definite plan, information necessities are mapped against the characterized techniques. To make a repeatable, reusable process for delivering strategies, the group needs to make a record that traces strategy plan guidelines. Just when these guidelines are pursued can the robotized program make genuine productivity and progress, by being repeatable and viable?

**Table 3.1:** Automation Procedure Design Process

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | **Architecture Review**. The group surveys the engineering so as to recognize the procedures that apply. |
| 2 | **Automation Step Definition (Development Level)**. A strategy definition is built at the advancement level, recognizing the methodology arrangement that applies for the different plan parts and procedures. |
| 3 | **Automation Step Definition (System Level)**. A method definition is built at the framework level, distinguishing the system arrangement that applies for the different procedures. |

|  |  |
| --- | --- |
| 4 | **Procedure Design Standards**. Structure measures are received and a one of a kind naming tradition is embraced that recognizes the methods on the task from techniques created previously or on different undertakings. |
| 5 | **Manual v/s Automated Tests.** Techniques is portrayed as being either performed physically or as a robotized test. |
| 6 | **Test Procedures Flagged for Detailed Design**. Test methods which emerge as progressively refined are hailed. These test methods at that point characterized as a major aspect of definite plan. |
| 7 | **Detailed Automation Design**. Those systems hailed as a major aspect of stage 7 are planned in further detail inside a point by point configuration record or report. Nitty gritty structure comprises of pseudo-code of calculations, fundamental test step definition, or pseudo-code of test automation programs. |
| 8 | **Data Mapping**. Technique lattice is changed to reflect test information prerequisites for each test methodology. |

The activity of building up the methodology definition helps being developed, as well as evaluates or bound the exertion. The improvement of the methodology definition includes the recognizable proof of the suite of techniques that should be produced and executed in help of the exertion. The plan practice includes the association of methodology into coherent gatherings and the meaning of a naming tradition for the suite of test systems.

At the framework level, it might be valuable to build up a nitty gritty plan for modern tests. This may include methods that perform complex calculations, comprise of both manual and mechanized advances, and test programming contents that are adjusted for use in different test methodology. The initial step is to survey the strategy definition at the framework test level. This audit is directed to identify those systems that emerge as being progressively complex and that, subsequently, are required to be characterized further as a major aspect of point by point test structure.

Definite plan may appear as program pseudo-code, when writing computer programs is required. The nitty gritty plan might be spoken to just as a grouping of steps that should be performed in help of an execution. When programming factors and various information esteems are included, the nitty gritty plan may mirror the programming build of a circle supporting an iterative arrangement of execution including distinctive qualities, together with a rundown or table recognizing the sorts of information or scopes of information required for the execution.

Following the execution of point by point plan, information prerequisites should be mapped against the characterized strategies. When information necessities are laid out, the group needs to design the methods for acquiring, creating, or building up the information.

The structure of the program (engineering) is regularly depicted in two different ways. One methodology association technique includes the legitimate gathering of strategies with the framework application structure parts, and is alluded to as a plan based engineering. Another strategy speaks to a system point of view and connects methodology with the different sorts of procedures spoke to inside the program show, and is alluded to as a strategy based engineering.

A comprehension of execution procedures is essential when creating structure and the program configuration models. Work force performing should be acquainted with the systems related with the white box and discovery test-approach strategies.

White box test methods are engaged at practicing programming program internals; discovery procedures analyze the application under test conduct against prerequisites that address testing by means of set up open interfaces, for example, the UI or the distributed API.

## 3.5.3 Automation Development

For automation be reusable, repeatable, and viable, improvement principles should be characterized and pursued. Subsequent to performing investigation and plan, the group is presently prepared to perform advancement.

Remember that the structure and improvement exercises pursue an iterative and steady methodology, so as to address the most noteworthy hazard usefulness in advance. Table 3.2 corresponds the improvement stages to the procedure stages. The mechanization procedures and steps plot in the table are deliberately lined up with the advancement procedure, and the execution of these means results in the refinement of strategies in the meantime wen programming engineers are making the product modules. Mechanized and additionally manual test methods are characterized amid the combination stage with the aim of reusing them amid the framework test stage.

**Table 3.2: Development/Test Relationship**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Development Process** | **Test Process** |
| Module (Unit) Development | Design module from requirements. | Perform test arranging and test condition setup. |
|  | Code module. | Create test plan and create test information. |
|  | Troubleshoot module. | Write contents or record situation utilizing module. |
|  | Unit test module. | Debug mechanized test content by running against module. Use devices that help unit testing. |
|  | Correction defects. | Rerun computerized content to relapse as deformities are adjusted. |
|  | Direct performance execution testing | Check that framework is versatile and will meet execution prerequisites. |
| Integration | Reconciliation test associated modules.  Audit inconvenience reports. | Combine unit contents and include new contents that exhibit module interconnectivity. Use device to help mechanized mix testing. |
|  | Right deformities and refresh imperfection status. | Rerun robotized content as a feature of relapse, as imperfections are adjusted. |
|  | Proceeded with execution testing activities | Check that framework is versatile and will meet execution prerequisites. |
| System Test | Review inconvenience reports | Integrate robotized contents into framework level strategies where conceivable, and build up extra framework level test techniques. Execute framework and record results. |
|  | Right imperfections and refresh deformity status. | Rerun mechanized content as a major aspect of relapse test as imperfections are redressed. |
| Acceptance Test | Review episode reports | Perform subset of framework testing as a component of exhibit of client acknowledgment test. |
|  | Correcting defects. | Rerun robotized content as a feature of relapse test as deformities are rectified. |

Numerous planning exercises need to occur before advancement can start. An improvement engineering is produced (portrayed in the following segment), which furnishes the group with a reasonable image of the advancement planning exercises or building squares fundamental for the effective formation of test strategies.

The group should tailor the example advancement engineering to mirror the needs of their specific venture. Some portion of these setup and readiness exercises includes the need to follow and oversee condition set up exercises, where material obtainments may have long lead times. Before the beginning of improvement, the group likewise needs to perform investigation to recognize the potential for reuse of existing techniques and contents inside the automation framework (reuse library).

The group needs to create methods as per advancement/execution plan. What's more, plan needs to dispense assets and reflect improvement due dates, among different elements. The group needs to screen improvement advancement and deliver advance status reports. Preceding the formation of a total suite of methodology, the group plays out a seclusion relationship examination.

The after effects of this examination help to fuse information conditions, plan for work process conditions among executions, and recognize basic contents that can be connected over and again to the exertion. As methods are being created, the group needs to guarantee that setup control is performed for the whole bed to incorporate structure, contents, and information, just as for every individual methodology. The bed should be baselined utilizing a design the board device.

Test, QA, Integration, Deployment advancement includes the improvement of systems that are viable, reusable, straightforward, and hearty, which in itself can be as trying as the advancement of the application under execution. Advancement norms ought to be set up supporting organized and predictable improvement of computerized execution. Improvement measures can be founded on the scripting dialect models of a specific apparatus.

The selection of existing framework improvement gauges is commonly a superior methodology than making a standard starting with no outside help. In the event that no advancement principles exist inside the association for the specific device scripting dialect, it's critical for the group to create content improvement rules. Such rules can incorporate bearings on setting autonomy, which tends to the specific place where a system should begin and where it should end. By creating strategies dependent on advancement rules, the group makes the underlying building obstructs for a mechanization framework.

The mechanization framework will store a library of normal, reusable contents. All through the exertion and in future discharges, the specialist can make utilization of the mechanization framework to help reuse of chronicled methods, limit duplication, and therefore upgrade the whole automation exertion.

## 3.5.4 Automation Development Architecture

Colleagues in charge of item advancement should be set up with the best possible materials. Colleague need to pursue a design that incorporates, for instance, a posting of the systems allotted to them and a posting of the result of computerized versus manual investigation. Additionally, group faculty need to choose when to robotize. Now and again a test group should need to abstain from computerizing utilizing a GUI device before the interface, regardless of whether API, character UI, or GUI, is balanced out, to abstain from having to reengineer the mechanization because of non– bug-related changes. At different occasions, the group may discover workaround arrangements while robotizing a temperamental GUI, for example, concentrating computerization on the known stable parts as it were.

The designer needs to hold fast to the technique advancement and execution plan, test structure data, mechanized instrument client manuals, and methodology improvement rules. Involved with the best possible directions, documentation, and rules, test architects will have the establishment that enables them to build up a progressively durable and organized arrangement of test systems. Rehashing a procedure and more than once showing a solid program relies upon the accessibility of reported procedures and standard rules, for example, the test improvement engineering.

## 3.5.5 Technical Environment

This movement should be upheld by a specialized domain, which encourages the improvement of methods. Accordingly, the earth should be set up and prepared to go. This condition incorporates the specialized condition, which may incorporate office assets just as the equipment and programming important to help test, advancement and execution. It should be guaranteed that there are sufficient workstations to help the whole group.

Condition setup exercises can likewise incorporate the utilization of a domain setup content to stack information or reestablish a drive picture, and to align the instrument to nature. At the point when instrument similarity issues emerge with the application under execution, workaround arrangements must be distinguished. When creating methods, it's vital that the calendar for creating techniques is predictable with the execution plan. It's additionally vital that the group pursue technique improvement rules.

The group must guarantee that the best possible test room or lab offices are held and set up. When the physical condition is set up, the group guarantees that all fundamental gear is introduced and operational. The arrangement characterized the required specialized condition and tended to condition arranging. Inside the earth area of the arrangement, the group has officially recognized operational help required to introduce and look at the operational availability of the specialized condition. The group needs to guarantee that operational help exercises have been legitimately booked and should screen advancement of these errands.

Explicit errands and potential issues delineated in the arrangement should now have been tended to and settled. Robotized instruments that apply ought to have been planned for establishment and checkout. These devices presently ought to be designed to help the group and be operational inside the test condition.

The earth setup movement incorporates the need to follow and oversee condition setup exercises, where material acquisitions may have long lead times. The hardware supporting the environment must be sufficient to ensure complete functionality of the production application. Environment hardware needs to support performance analysis. In cases where the environment utilizes hardware resources that are also supporting other development or management activities, special arrangements may be necessary during actual performance testing.

During system execution, the software configuration loaded within the execution environment must be a complete, fully integrated release with no patches and no disabled sections. The hardware configuration supporting the environment needs to be designed to support processing, storage, and retrieval activities, which may be performed across a local or wide area network, reflecting the target environment.

The environment design also needs to consider stress testing requirements. Stress and load tests may require the use of multiple workstations that will run multiple procedures simultaneously; some automated test tools include a virtual user simulation functionality that eliminates or minimizes the need for multiple workstations.

Data will need to be obtained with enough lead time to support refinement and manipulation to support requirements. Data preparation activities include the identification of conversion requirements, the preprocessing of raw files, loading of temporary tables, possibly in a relational database management system format, and the performance of consistency checks. Amid arranging, the group characterized and planned nature exercises.

Presently the group needs to follow the earth setup exercises. Assets should be recognized to introduce equipment, programming, and system assets into nature and coordinate and introduced condition assets. The earth materials and the application under framework should be baselined inside a setup the board apparatus. Also, execution condition materials may incorporate test information and test forms.

The group needs to get and alter databases important to practice programming applications, and create condition setup contents and proving ground contents. The group ought to perform item audits and approval of all source materials. The area of the earth for each venture or undertaking ought to be characterized inside the arrangement for each task.

# 3.6 Execution and Management of Automation

At this stage, the group has tended to test, structure and improvement. Condition setup arranging and execution was tended to steady with the prerequisites and rules gave inside the test plan.

With the test plan close by and the earth now operational, it's an ideal opportunity to execute the mechanization characterized for the test program. When executing systems, the group must agree to a methodology execution plan, as talked about prior. The technique execution plan actualizes the system characterized inside the test plan. Plans for unit, static examination, code inclusion mix, framework, and client acknowledgment testing are executed. Together, these execution stages make up the means that are required to execute the framework in general. The different advances required amid execution and the executives are delineated beneath.

* When executing automation systems, the group needs to conform to a strategy execution plan. Following execution, result assessments are performed and execution result documentation is readied.
* Plans for unit, incorporation, framework, and client acknowledgment testing are executed, which together make up the means that are required to official the framework all in all. With the unit testing, code profiling of programming can be performed. "Profiling is a tuning procedure that decides if a calculation is sufficiently proficient". Profiling additionally find occurrences where there is ill-advised scaling of calculations, instantiations, and asset usage.
* Integration testing centers around the application internals. In this, units are gradually incorporated and tried together dependent on control stream. Since units may comprise of different units, additionally called module testing, may happen amid unit testing.
* During framework test, the designer is trying the reconciliation of parts that involve the whole framework. A different group for the most part performs framework level executions. The group actualizes the method execution plan and the framework test plan.
* The group additionally performs investigation to distinguish explicit parts or usefulness that are encountering a more prominent relative number of issue reports. Because of this examination, extra test strategies and exertion may should be relegated to the segments. Results examination affirm whether executed execution methods are ended up being beneficial as far as recognizing blunders.
* Each group needs to perform issue announcing tasks in consistence with a characterized procedure. The documentation and following of framework issue reports is enormously encouraged by a mechanized imperfection following instruments and systems.
* The group supervisor is in charge of guaranteeing the execution as per timetable, and Team part are apportioned and diverted when important to deal with issues that emerge amid the execution exertion. To play out this oversight work successfully, the chief needs to perform program status following and the executives detailing.
* Metrics furnish the supervisor with key markers of the inclusion, advance, and the nature of the exertion. Amid white box testing, the specialist estimates the profundity of testing, by gathering information in respect to way inclusion and test inclusion. Amid black box practical testing, measurements gathering centers around the expansiveness of testing, to incorporate the measure of showed usefulness and the measure of execution that has been performed.

# 3.7 Automation Program Review and Assessment

Computerization program survey and appraisal exercises should be directed all through the item lifecycle, to take into consideration consistent enhancement exercises. All through the lifecycle and following execution exercises, measurements should be assessed and last survey and evaluation exercises should be led to take into consideration process enhancement? The different advances essential for program audit and appraisal are sketched out underneath.

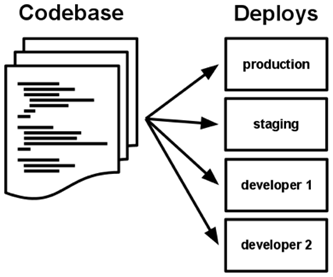
* Following execution, the group needs to audit the execution of the program to figure out where changes can be actualized to enhance the program execution on the following undertaking. This program audit speaks to the last period of the Automated Lifecycle Methodology (ALM).
* Throughout the program, the group gathered different measurements. The focal point of the program survey incorporates an evaluation of whether the application fulfills acknowledgment criteria and is prepared to go into creation.
* As part of its way of life, the group needs to embrace a continuous iterative procedure of exercises learned exercises. Such a program urges specialists to assume the liability to raise remedial activity recommendations quickly, when such activities conceivably have huge effect on program execution. All through the whole lifecycle, it's great practice to archive and start to assess exercises learned at every achievement. The measurements that are gathered all through the lifecycle and particularly amid the execution stage help pinpoint issues that should be tended to.
* Lessons learned, measurements assessments, and relating enhancement action or restorative activity should be archived all through the whole procedure in a focal storehouse that is effortlessly open.
* After gathering exercises educated and different measurements, and characterizing restorative activities, builds additionally need to survey the viability of the program to incorporate an assessment of the program rate of profitability. Designers catch proportions of the advantages of computerization acknowledged all through the lifecycle so as to help this evaluation.
* Teams can perform studies to ask about the potential estimation of process and instrument necessities. An overview frame is utilized to request criticism on the potential utilization of necessity apparatuses, plan devices, and advancement instruments.

# 3.8 Factors to Make App Infrastructure Agnostic

The 12 Factor App requirements make sure that the app can be made infrastructure agnostic, be it cloud or standalone. From an Engineering perspective, we have to make sure the below standards are maintained and the arrangements of the infrastructure/resources are as per the twelve factor requirements below:

## 3.8.1 Codebase

One codebase tracked in revision control, many deploys*.* Each app should have only one code base which is version tracked with proper version maintenance of the code. One App cannot have different code bases. There can be different deploys of the same app running. Deploys are running instances of the same app. Each deploy can have different version of code depending upon the code build pipeline status.



**Figure 3.1: Codebase Deployments**

## 3.8.2 Dependencies

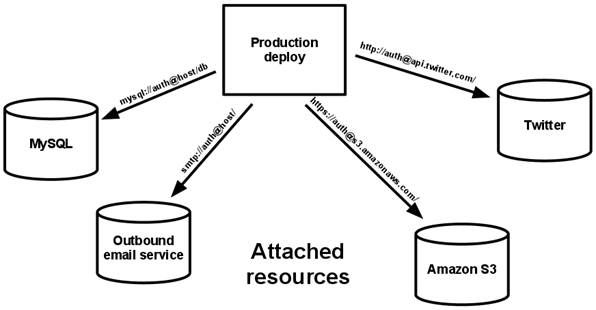
Explicitly declare and isolate dependencies. The dependencies should not be tied implicitly inside app or classes and should be declared in one place for the whole app. Any new developer looking at the app should be able to just pull in dependencies from the dependency management tool and not worry much about pulling in implicit dependencies and should be able to start building and running the app within no time.

## 3.8.3 Configuration

Store configuration in the environment. Each environment (deploys) should have its own specific variables declared in an environment specific configured files. The easy test of it is that the app should be able to be made open source anytime for its configuration to be kept separated.

## 3.8.4 Backing Services

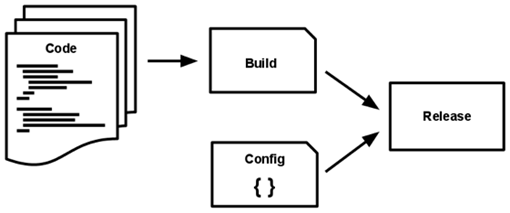
Treat backing services as attached resources*.* All the resources whether local DBs or external services should be loosely coupled and can be replaced easily in future if we want to. The backing services should be easily configurable and can be easily replaced in case of failures.



**Figure 3.2: Backing Services**

## 3.8.5 Build, Release, Run

Strictly separate build and run stages. The build stage for an app should be separate and be executed before the run(release) stage to make sure the build doesn't have any issue before running the app itself. This helps to isolate the problem and track it easily to fix any issues in lesser time.



**Figure 3.3: Build Release Run**

## 3.8.6 Processes

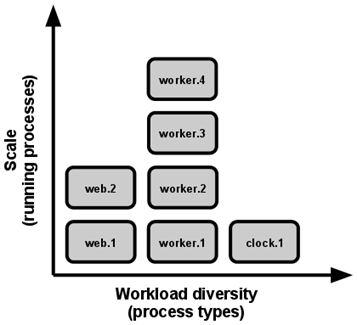
Run the application as stateless processes. Everything should be stored in a data store and any future processes should not depend on the cached data on filesystem or any place. Each process should be independent on its own and should not depend on data of any other process' state.

## 3.8.7 Port Binding

Export services via port binding. Expose apps via URL: port for easy access to other external or internal applications. Each app and its services should be easily accessible via URL endpoint whether it is used by an external or internal API or application.

## 3.8.8 Concurrency

Scale out via the process model. Each app should have its own process types/groups defined like below to scale out the processes as per the requirements in that particular process type. Array of process types and number of processes of particular type is known as the process formation. App should not demonize or write PID files in system, instead it should let the OS handle the processes spun out from the APP which can be distributed over different systems as per the process loads.



**Figure 3.4: Workload Diversity**

## 3.8.9 Disposability

Maximize robustness with fast start-up and graceful shutdown. *Processes should be able to start up and shutdown in small amount of time and should be able to resume the program execution in minimal amount of time. The system should be robust enough to handle such failures and get back online in no matter of time.*

## 3.8.10 Dev/Prod Parity

Development, staging, and production should be as similar as possible. *There should be minimal gaps in processes to take app from development to production. The different gaps which need to be minimized are as below:*

1. The time gap: A developer may work on code that takes days, weeks, or even months to go into production.
2. The personnel gap: Developers write code, ops engineers deploy it.
3. The tools gap: Developers may be using a stack like Nginx, SQLite, and OS X, while the production deploy uses Apache, MySQL, and Linux.

## 3.8.11 Logs

Treat logs as event streams. *A 12 factor system should not worry* about storing log files, it should just be writing all its logging events to standard output and let the deployment infrastructure take care of handling logs. The log are events of things happening in the app sorted in a timely manner which need to be routed to appropriate tools to handle logs for analysis and alerting.

## 3.8.12 Admin Processes

Run admin/management tasks as one-off processes. All the admin related tasks whether it is in development or production should be executed as on-off processes which once triggered should handle all the admin related task in one go than executing multiple processes for the same.

Depending upon the organization structure and priority, the above factors should be incorporated in an application. Below are some the priorities currently in place for the Cloud Native Architecture requirements:

**Table 3.3: Admin Process Priority and Description**

High = Cloud ready, Medium = Cloud friendly, Low = Cloud Resilient

|  |  |  |
| --- | --- | --- |
| Factor | Priority | Description |
| 1. Codebase | High | One code base tracked in revision control, many deploys |
| 2. Dependencies | High | Explicitly declare and isolate dependencies |
| 3. Configuration | Medium | Store configuration in the environment |
| 4. Backing Services | Medium | Treat backing services as attached resources |
| 5. Build, Release, Run | Low | Strictly separate build and run stages |
| 6. Processes | High | Execute the app as one or more stateless processes |
| 7. Port Binding | Low | Export services via port binding |
| 8. Concurrency | High | Scale out via the process model |
| 9. Disposability | Medium | Maximize robustness with fast start-up and graceful shutdown |
| 10. Dev/Prod Parity | Medium | Keep development, staging, and production as similar as possible |
| 11. Logs | High | Treat logs as event streams |
| 12. Admin Processes | Low | Run admin/management tasks as one-off processes |

**References:**

Alok Mani Tripathi (Author), Learning Robotic Process Automation Paperback – Import, 1 Mar 2018

Jez Humble (Author), David Farley (Author), Martin Fowler (Foreword), Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation Textbook Binding – 2016

Jez Humble (Author), David Farley (Author), Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series) Hardcover – 27 Jul 2010

Mark Fewster and Dorothy Graham, Software Test Automation: Effective Use of Test Execution Tools.

Robert C. Martin (Author), Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin) Paperback – 25 Sep 2017

https://en.wikipedia.org/wiki/Automation

https://en.wikipedia.org/wiki/Test\_automation

https://en.wikipedia.org/wiki/Cloud\_storage

https://en.wikipedia.org/wiki/Software\_testing

https://en.wikipedia.org/wiki/Software\_release\_life\_cycle