International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 8, Issue 12, December-2023, http://ijmec.com/, ISSN: 2456-4265

WIRELESS INTELLIGENT VIDEO SURVEILLANCE SYSTEM USING MOVING OBJECT RECOGNITION TECHNOLOGY

Dr. Bhargav Gangadhara,
[1]Senior Technical lead, Jack Henry and Associates, *USA*
[1] *bhargavmtechmem@gmail.com,*

Abstract— This paper presents an innovative approach to cycle time reduction in internal quality management through the implementation of an automated tool. The focus is on optimizing resource planning for product-based organizations by analyzing variables such as resource numbers, request types, and testing methods. Statistical modeling is utilized to enhance work allocation and provide precise status updates to development leads, ultimately leading to significant cost savings and efficiency improvements.

The developed automation tool, as an alternative to manual test case testing, utilizes statistical modeling outputs to assist in work allocation to members of testing team and also to collect the actual time spent on each of the activities performed by them for further analysis. This will also help to provide correct status to development leads. So that results can be communicated to development leads in advance whether their request can be met or not. Resource planning is also required to do random testing in case there are few requests or quality of the product is critical and this will help in identifying how much testing is done.

The next task after resources planning is server upload, which takes a lot of bandwidth of testing group to upload the document and source code. Optimization of server based on understanding of upload sequencing and server load. To meet the current load which will save several million dollars as the server cost is high.

Keywords- *Automation, Co-relation, Testing Statistical modeling, Resource planning, server upload, Upload sequencing, Bandwidth.*

I. INTRODUCTION

As a software quality assurance percipient, the major concern was to identify a critical area for research on internal quality management. Even though concept seems to be simple on papers practically performing takes some pain, let us understand the organization structure. The organization is into business of providing the solution to mobile phones. This consists of development team, internal testing team and validation team. The product may also undergo test from external source (i.e. factory quality control) before delivering the product to the consumer. As the consumer satisfaction and the company image is the major concern for any product development company. The development team is into research of new applications for android mobile phone. Also, other lower tier models are manufactured with lesser cost accordingly. since the country adaptation is different for different countries. The quality has to be maintained as chinese customers may finally end up in getting mobile hand set with applications supporting Indian languages. Intended software for correct user, functional and non-functional working condition of the mobile hand set is tested manually, which may cause errors so automation plays a vital role for product development and release to factory.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

The purpose of Development Life cycle model guidelines is to help projects deploy the standard lifecycle model in the Software Department projects. The organization has defined the life cycle to be followed in the Software projects as below. This lifecycle has been named as "Software Development Lifecycle (SDLC)" and is found to be more prevalent among the projects in the Software department.

The following are the reasons for defining the Solution Department specific life cycle:

- Customer requirements: Customer would like to have intermediate milestones to check the product development progress. Hence all the projects are broken down into many milestones. The number of milestones in a project depends on the Complexity, size and project dependencies (availability of hardware etc).
- Lessons learnt: Prior to 2009, projects used to follow pure waterfall life cycle model for development. This created uncertainties in project till the product is delivered. As a risk mitigation strategy, organizations software group decided to use its own life cycle and the following model was defined.
- Evolving scope: Customer could evolve the scope of project after reviewing the progress during intermediate milestones. This provides more flexibility to customer and better service.

1.1. Software Development Lifecycle

A project is divided into multiple milestones. Each milestone needs to result in a working deliverable of Project and spanning between 3 to 4 months of duration.

The following is the architecture of milestone management of SDLC process model

- **M0:** This basically constitute the high level system architecture and high level requirement analysis of the entire project. But, it would not involve any activities of deriving the detailed requirements of the subsequent milestones.

- **M1, M2, M3…:** This involves execution of all life cycle phases of SDLC (Requirement, Design, Coding, Unit Testing, Integration Testing, and System Testing). The detailed requirements of each of these milestones are derived as part of respective milestone only, by referring the high level architecture derived in M0.

Note: Milestones can be executed either sequentially or in parallel based on project requirement. Some of the combinations are listed below.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

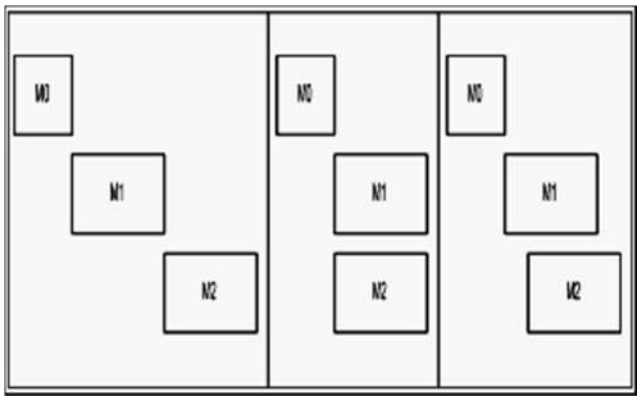


Fig 1: Milestone management for software development life cycle

The development life cycle is shown below

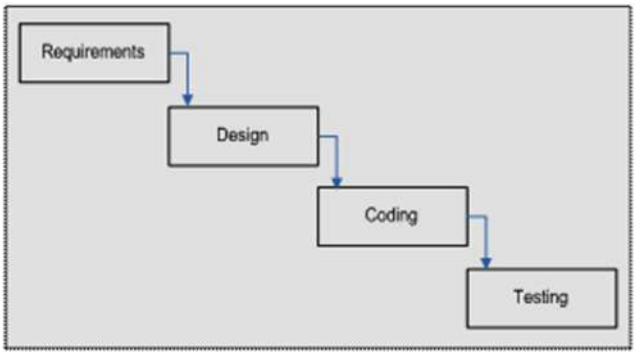


Fig 2: Software development life cycle

1.2. Phases

Phase and Process areas are distinct and should not be used interchangeably. The activities performed within the phases will be governed by Processes. For example, activities such as Unit Test planning and Unit Test performed within the Coding phase are governed by the Unit Test Process.

The following defines the scope of the various Phases:

- Requirements Phase: This phase consists of the following activities:
  - Requirements development
  - Requirements planning and allocation
  - Requirements reviews
  - Requirements rework

- Design Phase: This phase consists of the following activities:
  - Design
  - Design reviews
  - Design rework

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

- Coding Phase:
  - Coding, Code reviews, Code rework, Unit test planning
  - Unit test preparation including test case generation
  - Unit testing
  - Integration planning, Integration, Integration test planning, Integration test preparation including test case generation, Integration testing
  - Unit testing and Integration testing related reviews and rework

- Testing Phase:
  - Test planning
  - Test preparation including test case generation
  - Acceptance test scenario generation
  - System testing and Acceptance testing related review and rework
  - System testing
  - Acceptance testing

The testing group has majorly two tasks, one is testing and the other is to release to downstream user. The team is provided with finite number of resources. But there's number of request from development team, which is variable and does not follow any pattern.

II. INTERNAL FLOW OF DATA IN ORGANIZATION

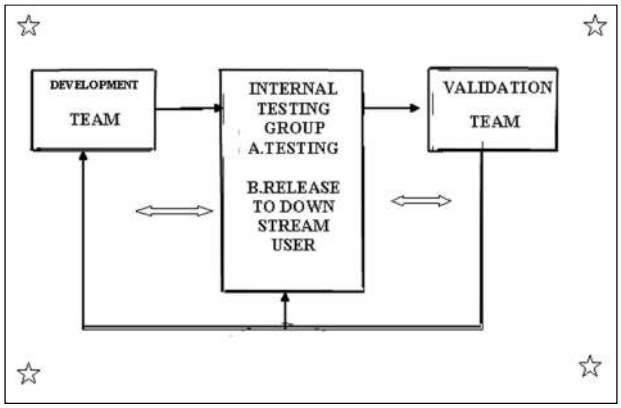


Fig 3: Internal organization structure

2.1. Task

The development team release information regarding the time taken for start and end of a model development.

The internal testing group perform the various types of tests, here its sanity test and release the output to downstream users.

The validation team then validates the tests performed and certain control measures are specified as feedback to development leads.

DOI: https://doi-ds.org/doilink/12.2023-39162716

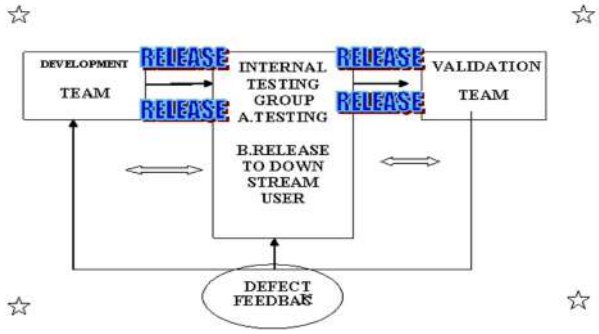


Fig 4: Internal flow of data in organization

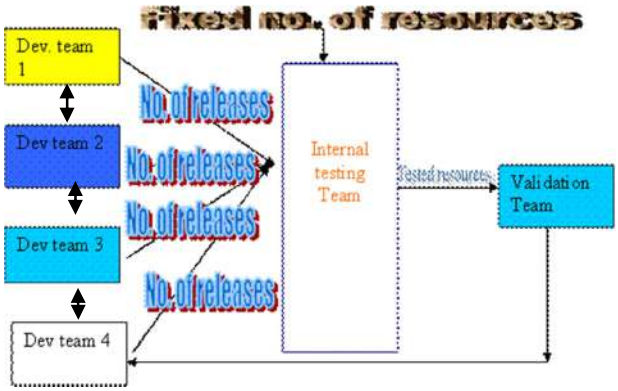


Fig 5: Real time release of data to various teams.

III. PROBLEM DEFINION AND ROOT CAUSES

The resources available from the development team will sent to the internal testing team as explained earlier , the risk involved in process is to be analyzed and the root cause for the risk is to be found. The Fig 7 explains that the maximum risk involved is due to resources which are not efficiently utilized by the internal organization.

Here we can clearly see that resource allocation has the maximum risk counts and is causing the problem.

The root cause for why the risk cont is high during resource allocation needs analyzed. After in-depth analysis of the organizations data release to internal testing team and verification team, human resources, manual testing of test cases, effort required for testing each test cases, effect of manual testing on productivity.

I found that two such cases which correspond to each other where in solving one problem may act as a solution to other problem.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

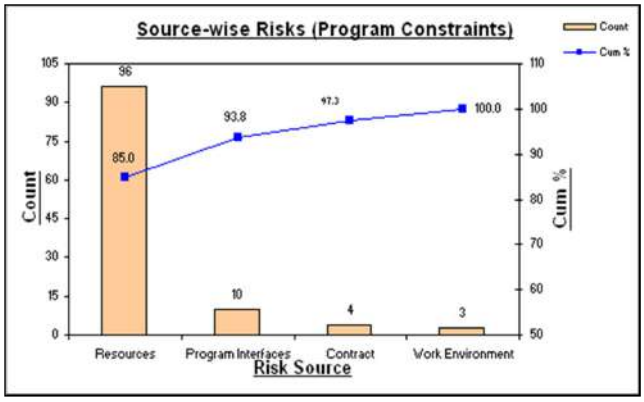


Fig 6:Source-wise Risks (Program constraint)

Fig 7: Pie-chart of Risk Source

Root cause 1: manual testing for test cases.

Root cause2: upload sequencing, server load and analysis of bottle necks in networks.

Preventive actions

Root cause 1: automation tool instead of using manual testing

Root cause 2: optimize the overall cycle time of server based on understanding of upload sequencing, server load and analysis of bottle necks in networks.

Here its simple to understand as the test cases are automated the result of the test cases are know at a particular time .Example: If the testing engineer runs the automation tool say at morning 9 am and expects the completion of all the test cases at 12 pm. As the tool is automated result is fixed and time management is easy .Now resource planning is also simple as release the data related to model defects can be intimated to respective development leads.

The synchronization of release defect data to development leads was a major concern , now particular time is been set to release the defect data .Example: If the result of testing was out at 12pm for model X , the test cases which have failed is noted and conveyed to the particular model development leads. So that the release of failure

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

data related to model X is scheduled at 12:05pm with 5minutes delay time. Similarly this can be useful if the number of models released during a day is more than 10 to 15 for different country adaptation.

IV. MANUAL V/S AUTOMATED.

Nowadays manually executing the processes takes more time and the human resources required for the verification of test cases is humongous. Even though the internal testing team works in different shifts , manual deployment of test cases takes more time the remedy for such bottle necks is to automate a tool which helps the testing of functional like sanity testing and non-functional like setting alarm, deleting messages, stress checking the contact capacity which may be limited to 250 contacts only. Here the automation is essential in non-functional test cases where the test engineer can test certain applications such as stress checking any application for more the one day ,even if he's not in working place he can run the automation tool which helps him to stress check that particular application and if there is any critical issue related to test cases then the development lead can be intimated which helps the development leads to know the error well in advance and control measures can be taken to overcome this problems. But in case of manual testing for functional applications like the sanity test companies prefer manual testing itself as knowledge of different domain is essential for a tester which cannot be updated in an automation tool. This untouched area wherein manual testing was essential, an automation tool is developed which reduces the burden of the tester and also manual vigilance is applicable.

Here we use resource planning and cycle time reduction in testing group. The types of test performed are Sanity Testing or detailed testing...etc. is a brief test of major functional elements of a piece of software to determine if it's basically operational. Sanity testing is used to verify that the software build is ready for System Testing (or more extensive testing). The tests target the most frequently used functional areas, key functions & newly implemented functions for the particular build version.

If an application does not pass the Sanity Test, then further testing activities are suspended.

The conditions which get checked in this test is general, like existence of application, loads without any problem, does not terminate abruptly, connects to correct database, all menu options are displayed correctly and the start form / page / window is loaded correctly.

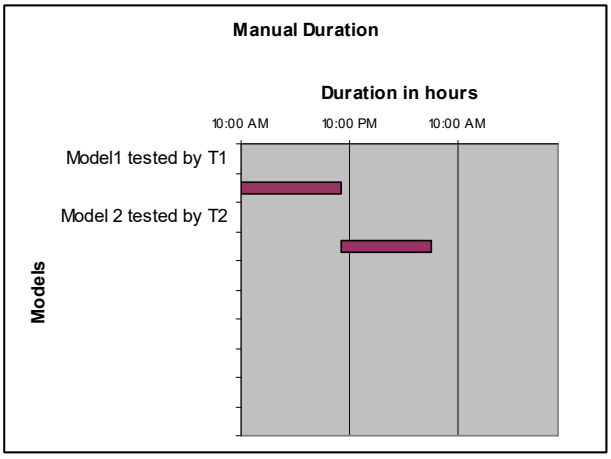


Fig 8: Gantt chart for manual testing

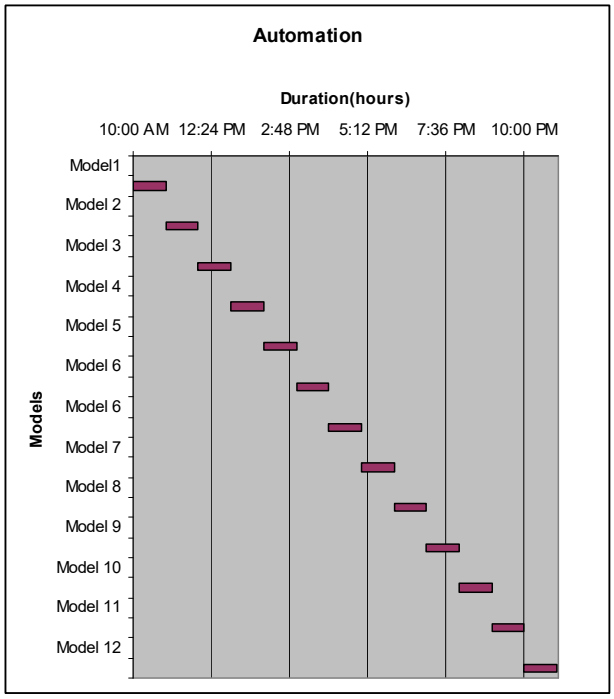


Fig 9: Gantt chart for Automation

V. AUTOMATION TOOL

5.1. Upload sequencing and server load.

Finally, the next task after automation tool for test cases is to release to downstream users, which takes a lot of bandwidth of testing group to upload the document and source code. Optimization of server based on understanding of upload sequencing and server load. To meet the current load which will save several million dollars as the server cost is high.

Resource planning is also required to do random testing or ad hoc testing in test case (e.g. Sanity testing, Unit testing, integration testing, and system testing and acceptance testing).random testing or ad-hoc testing is done by test engineers wherein they want to test applications such as word dictionary in mobile phones, after all the major testing is done suddenly conducting an ad-hoc test in case there are few requests or quality of the product is critical will improve the quality of the applications in the mobile phone .In turn the customer is satisfied.

VI. DATA COLLECTION

6.1. Raw data

The data collected from March to October from the respective development leads and with the help of simular software and also manually analyzed the given metrics, also prepared a line chart for number of models-months taken for developing particular models.

The development of the models and duration for development is also verified and for better understanding line charts where created as shown in Fig 8.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

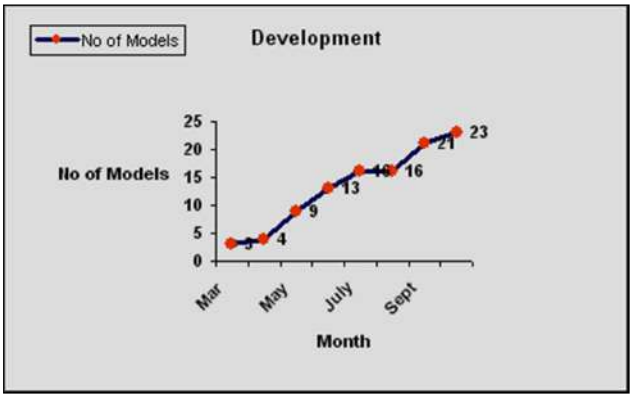


Fig 10: Line chart for development No. of Models-months

The Deployment of test cases data were also collected from model leads and analyzed the variance in the particular data per month. Sanity test which is for functional application such as key press, key map was collected and the time taken for each test conducted are shown below in Table 2. As we know that the functioning of keys in a mobile phone is a major criterion so the testing team usually conducts such tests frequently. Here the test leads conduct sanity test and the time management system which is an internal system in the organization calculate the time taken for each test and also time taken for each activity is also noted such as key map, key press and any special applications such as word dictionary. The Table 2 clearly shows the test conducted and the time taken for each month. After the further analysis of data for future upcoming months

6.2. ANALYSIS AND CONTROL

The ranges given in the acceptance line can be used to determine whether the performance of review falls within acceptable limits. To ensure that the performance of a review is checked with respect to control limits after each review takes place , this check is specified with respect to the control limits after each review takes place, this check is specified as an exit criterion for the review process Although the exit criteria can be defined as in-range checking of all of the various parameters, because defects is the central purpose of reviews, the exit criterion is the overall defect density should lie within the specified limits (Checking that the defects densities for the two types of defects are within the appropriate ranges can also be used.)If the number of defects found during the review is within the range given in the acceptance line, the review is considered effective, the exit criteria are satisfied, and no further action is needed for this review.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

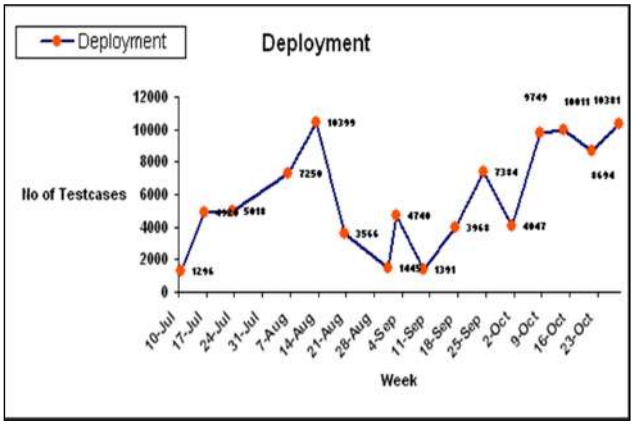


Fig 11: Deployment of No. of test cases-week

If the density of defects found in a review is not within the range given in the capability acceptance line, then the exit criteria are not satisfied. In software, however, the fact that the performance has gone out of the specified range does not automatically mean process failure. Instead, the moderator has to critically evaluate the situation and decide on the next steps. The preparation rate and review rate becomes very useful here—if the review rate is "too fast" as compared with that given in the acceptance line, then the reason for minor and critical defects can also be useful in this analysis. When the critical process is recognized, we try to find out the root cause of the critical issue as explained earlier.

To solve the first root cause in a process

The comparison of manual deployment of test cases like the functional and non-functional test cases which is comp aired with the automated tool evaluated test cases. Here we can see a drastic reduction in time e.g. - if a manually tested test case takes one hour, then the automated tool tested test case takes three minutes, it's an astonishing 57 minutes difference between the manual and automated test case.

Finally, to solve the problem relating to release to downstream user which caused serve down issues, it is necessary to understand the co-relation between the variables like Number of resources, Number of request and Type of testing.

Statistical modeling will help to optimize the resource planning. As shown in the Fig 11 if the resource planning and cycle time reduction is efficiently carried on during loop 1 itself then any critical error caused while development is negotiated then and there itself ,which saves time and cost incurred for retesting the product.

VII. DIFFICULTIES FACED DURING MANUAL DEPLOYMENT

- *Available resources may be different*
- *Speed of release is reduced as manually verifying the test cases may take some.*
- *Volume of the testcases constant and verification is bulk work.*
- *Late night releases which causes leads to stay for more than expected time.*
- Cost for both human resources and release .

Month	Sanity test	Key Map	Key press	Review
July	1032	491	435	9276
Aug	516	158	7711	14275
Sep	3928	327	2911	10317
Oct	24900	3233	0	14749

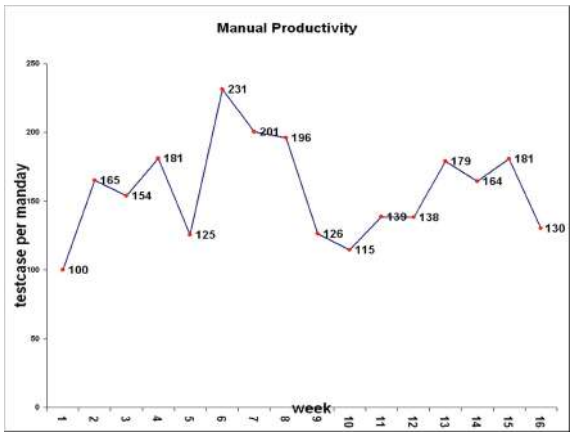


Fig 12: Manual Productivity

- ***Extra computer is required for manual testing.***

VIII. BENEFITS OF AUTOMATION

- *Extra computer for testing perpose is not required*
- *Automation may increase the speed of releases to down stream users.*
- *Voluminous jobs can be distributed.*
- *Cost saving.*
- *Accuracy is maintained.*
- *Errors caused during manyual testing is reduced*
- *Test engineers have their work cutoff.*
- *Stress testing is possible as the test engineers may run the automation tool for several days .*

DOI: https://doi-ds.org/doilink/12.2023-39162716

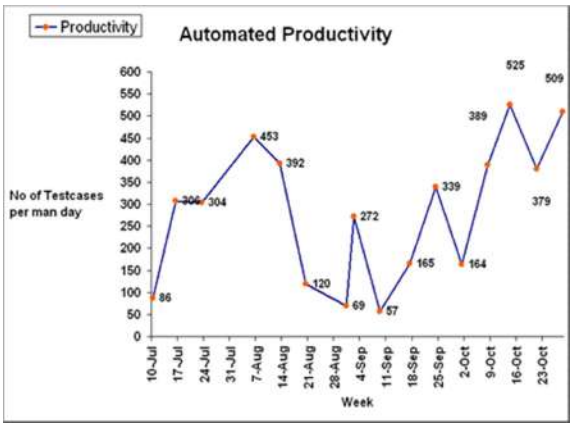


Fig 13: Increase in productivity per due to automation

IX. CYCLE TIME REDUCTION

Cycle time is the time required to complete a given process. The cycle time required to process a customer order might start with the customer phone call and end with the order being shipped in this example. The overall process is made up of many sub-processes such as order entry, assembly, inspection, packaging, and shipping. The cumulative cycle time of all of the sub-processes in your operation determines when you can promise product to your customer.

Benefits of cycle time reduction?

- Reduced costs
- Streamlined processes
- Improved communications
- Improved on-time delivery
- Improved productivity

How can Automation tool help implement a Cycle Time Reduction plan?

Automation tool works with you and your team to understand your business and its processes. Together, we map out your company processes, investigating new methods and identifying opportunities for reducing non-value added activity. Once this is complete, the field engineer helps you create and implement a plan for cycle time reduction.

ISSN: 2456-4265

DOI: https://doi-ds.org/doilink/12.2023-39162716

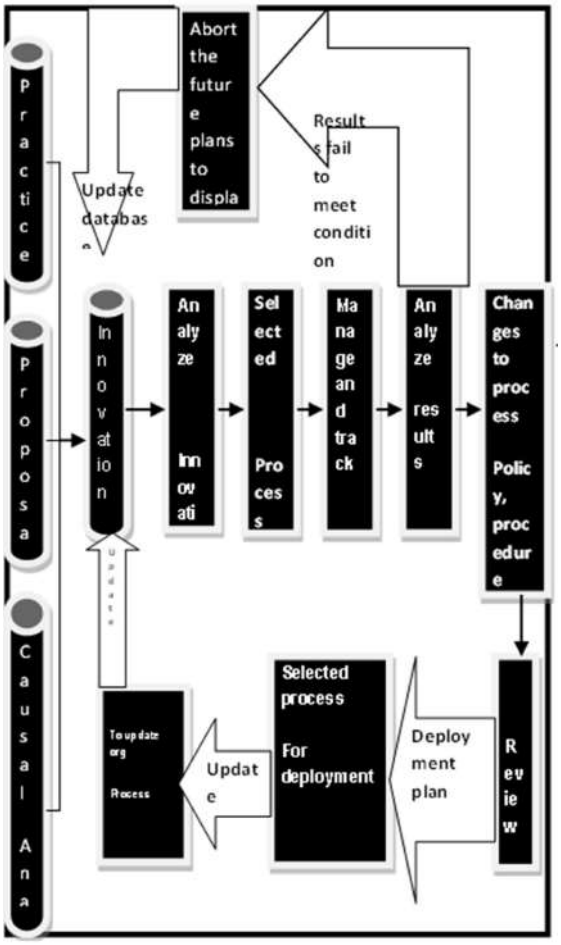
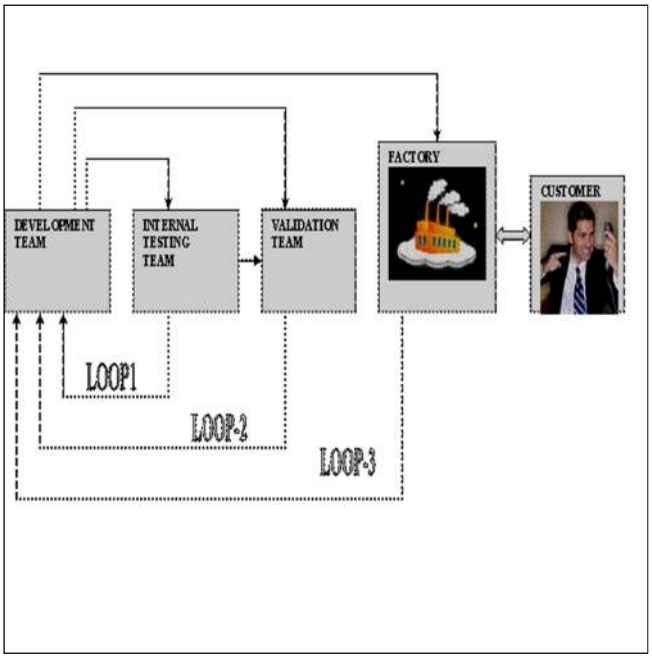


Figure 14: Process flow for automating a tool



DOI: https://doi-ds.org/doilink/12.2023-39162716

Fig 15: Cycle time reduction.

As shown in Fig.15 if the defects are verified well in advance in loop 1 itself the errors carried on to further processes is reduced.The development team leads and also the test leads are intimated with the errors.so that the leads can take up control measures to overcome errors.

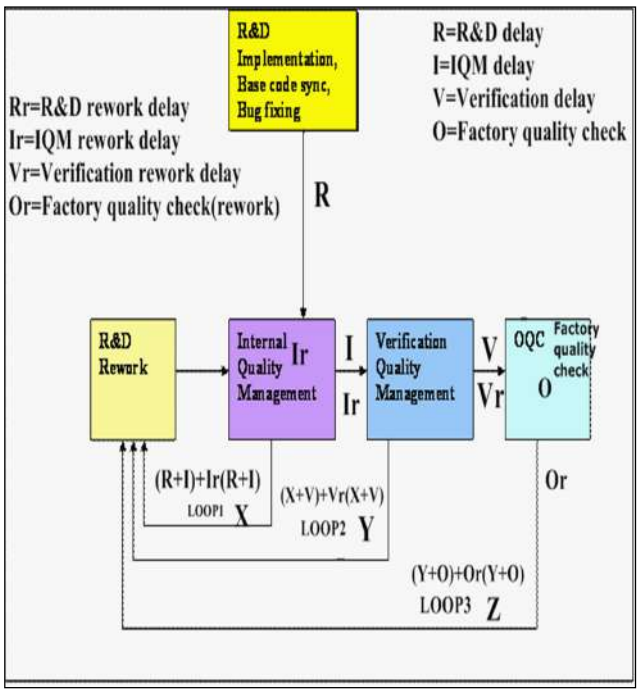


Fig 16: Calculation of cycle time

The above figure explains about the time taken for rework if the defects are carried on to next stages .Its obvious that the cost and human resources required for the rework or the defect correction is humongous .The calculation part is explained below .

Table 1:Banaries for calculating cycle time.

Rework	Internal quality management	Verification Quality Management	Factory Quality management
Ir	1	0	0
Vr	0	1	0
Or	0	0	1

(R+I)+Ir(R+I)—(X)

(X+V)+Vr(X+V)—(Y)

(Y+O)+Or(Y+O)—(Z)

X. CONCLUSION

DOI: https://doi-ds.org/doilink/12.2023-39162716

The study conclusively demonstrates the efficacy of automation tools in reducing cycle time for internal quality management. The implementation of these tools not only streamlines the testing process but also contributes to substantial cost savings and improved resource management, thereby enhancing overall product quality and ensuring customer satisfaction.

In the FIG.15 if the resource planning and cycle time reduction is efficiently carried on during loop 1 itself then any critical error caused while development leads are negotiated then and there itself ,which saves time and cost incurred for retesting the product. An automation tool can help human resource and extra computer required for testing, voluminous jobs can be distributed, speed of testing is increased and finally overall cost is reduced .The second issue which helps in scheduling the release to downstream users which causes server down scenario which can be reduced, the test leads as well as the development leads will know when to release the data , also to predict and optimize overall cycle time of server based on understanding of upload sequencing ,server load and analysis of bottle necks in network. To meet the current load a new server has to be installed which is costly, optimization of cycle time will handle this issue efficiently .

REFERENCES

[1] Dr.Bhargav Gangadhara , “Optimize Utility in Computing-Based Manufacturing Systems Using Service Models and Development Models”, Presented and published at International Journal of Computer Science and Information Security, IJCSIS Editorial Board, Vol -14, Page 5.

[2] Dr.Bhargav Gangadhara,”Optimize utility in cloud-Based manufacturing systems using service models and development models”, Presented and published at International Journal of Innovative Research in Advanced Engineering.IJIRAE Editorial Board, August 11 - 12, 2016.

[3] Dr. Bhargav Gangadhara, "Optimizing Cloud - Based Manufacturing: A Study on Service and Development Models", International Journal of Science and Research (IJSR), Volume 12 Issue 6, June 2023, pp. 2487-2491, https://www.ijsr.net/getabstract.php?paperid=SR23626155823

[4] K.Sakamoto, “Towars computational support for software process improvement activities,”Proceedings of the 20th International Conference on software engineering, pp. 21–31, 1998.

[5] M.Paulk, “The Capability Maturity Model for software :Guide lines for improving software process”,Addison-Wesley,1993.

[6] V.R.Basili and D.M.Weiss. “Evaluation of a software requirements document by analsis of change data.”Proceedings of the 5th international Conference on Software Engineering, pp. 314-323,1981

[7] W.Humphrey.”Managing the Software Process.”Addison-wesley,1989.

[8] B.W.Boehm.”Software Engineering economics.”Prentice Hall,1981.

[9] D.C.Montgomery.”Introduction to statistical Quality Control,”third edition.John Wiley and Sons,1996.

[10] A.J.Albrecht and J.R.Gaffney.”Software function, source lines of code and development efforts prediction:a software science validation.”IEEE Transaction on Software Engineering,9(6):639-648,1983.

[11] J.D.Musa, A.Iannino, and K.Okumoto.”Software Reliability – measurement,Prediction,Application.”McGraw Hill,1987.

[12] R.N.Charette.”Large-scale project management is risk management”,IEEE Software,pp.110-117,July 1996.

DOI: https://doi-ds.org/doilink/12.2023-39162716

[13] R.G.Ebenau and S.HStrauss."Software inspection process."McGraw Hill,1993.

[14] D.H.Kitson and S.M.Masters."An analysis of SEI software process assesment results:1987-1991.Proceedings of the 15th international conference on software Engineering,pp.68-77,1993.

[15] D.P.youll."Marketing software Development Visible—Effective Project control."John Wiley and sons,1990.

DOI: https://doi-ds.org/doilink/12.2023-39162716