

**IMPLEMENTASI *ROBOTIC PROCESS AUTOMATION*
PADA PROSES REKAPITULASI *INVOICE*
STUDI KASUS : PT. SCHNEIDER ELECTRIC
MANUFACTURING BATAM**

PROPOSAL TUGAS AKHIR

Oleh:

Adrian Priabdi Fauzi 4311711001

Disusun untuk pengajuan proposal Tugas Akhir Program Diploma IV



**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2020**

HALAMAN PENGESAHAN PROPOSAL

**IMPLEMENTASI *ROBOTIC PROCESS AUTOMATION*
PADA PROSES REKAPITULASI *INVOICE*
STUDI KASUS : PT. SCHNEIDER ELECTRIC
MANUFACTURING BATAM**

Oleh:

Adrian Priabdi Fauzi 4311711001

Proposal ini telah dikonsultasikan dengan dosen pembimbing sebagai persyaratan
untuk melaksanakan Sidang Proposal
pada

**PROGRAM DIPLOMA IV
PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
Batam, 2020**

Disetujui oleh:
Pembimbing I

Maidel Fani, S.Pd., M.Kom.

NIK. 117192

Abstrak

PT. Schneider Electric Manufacturing Batam merupakan salah satu industri yang telah melakukan transformasi digital dengan berbagai solusi digital, salah satunya adalah digitalisasi rekapitulasi *invoice* untuk merekap *invoice* dengan menerapkan teknologi *Robotic Process Automation*. *Robotic Process Automation* yaitu perangkat lunak mampu menirukan aktivitas yang dilakukan oleh manusia dalam memproses atau memasukkan data secara otomatis dalam sistem komputer, sehingga dapat mengurangi tingkat kesalahan pada proses rekapitulasi dan waktu proses dibutuhkan menjadi lebih sedikit.

Penelitian ini akan meriset dan mengembangkan aplikasi menggunakan teknologi *robotic processing automation* untuk proses rekapitulasi *invoice* dengan menggunakan metode *software development life cycle waterfall* serta menguji tingkat efektivitas penggunaan *robotic processing automation* dalam proses rekapitulasi *invoice* menggunakan gabungan metode pengujian *white-box* dan *black-box*.

Hasil penelitian ini menunjukkan bahwa pengembangan aplikasi *robotic processing automation* dapat menggunakan metode *software development life cycle waterfall*. Kemudian *robotic processing automation* dapat digunakan pada proses untuk menyerap informasi, memvalidasi, dan merekapitulasi dokumen dengan tingkat efektivitas yang bagus berdasarkan hasil rekapitulasi dokumen *invoice*. Berdasarkan hasil tersebut dapat disimpulkan bahwa pengembangan aplikasi dengan teknologi *robotic processing automation* dapat menggunakan metode *software development life cycle waterfall* dan penggunaan *robotic processing automation* dalam proses rekapitulasi *invoice* memiliki tingkat efektivitas yang bagus.

Kata Kunci : *RPA, Invoice, Rekapitulasi*

1 Latar Belakang

PT. Schneider Electric Manufacturing Batam merupakan salah satu industri yang telah melakukan transformasi digital dan menjadi *smart factory program* pada tahun 2017, mendapatkan pengakuan sebagai *Industrial Revolution (4IR) Lighthouse* oleh *World Economic Forum (WEF)* dan *Lighthouse For The Implementation Of Industry 4.0 In Indonesia* oleh *Making Indonesia 4.0 Lighthouse*. *Lighthouse* merupakan perusahaan percontohan dalam implementasi industri 4.0, sedangkan *smart factory* merupakan bagian dari transformasi digital oleh Schneider Electric secara Global. Berbagai solusi digital telah dihasilkan dari perjalanan digital transformation Schneider Electric seperti, *remote glass assistance*, *augmented operator advisor*, *virtual reality*, *digital inspection/outgoing insight* dan masih banyak solusi – solusi digital.

PT. Schneider Electric Manufacturing Batam terdapat salah satu proses bisnis yaitu proses rekapitulasi *invoice*, yaitu proses yang dilakukan untuk merekap dan memvalidasi *invoice* yang masuk dari vendor. Proses tersebut bertujuan untuk memeriksa dan memvalidasi apakah *invoice* yang diterima dari vendor sudah pernah terima atau terjadi duplikasi. Pada saat ini, proses rekapitulasi *invoice* masih dilakukan secara manual yaitu melibatkan karyawan dalam memeriksa dan memvalidasi informasi yang terdapat dalam *invoice*. Kemudian karyawan akan mengirim informasi kepada bagian *finance* untuk melaporkan hasil dari rekapitulasi *invoice*. Proses tersebut dilakukan secara terus menerus, sehingga membutuhkan waktu yang lama, sumber daya manusia yang banyak, dan resiko terjadi kesalahan menjadi tinggi, sehingga kemungkinan terjadi kesalahan dalam hasil rekapitulasi *invoice* juga akan tinggi.

Pada proses rekapitulasi *invoice* ini, PT. Schneider Electric Manufacturing Batam akan menerapkan salah satu teknologi automisasi bernama *Robotic Process Automation*. *Robotic Process Automation* atau *RPA* adalah perangkat lunak yang mampu menirukan aktivitas yang dilakukan oleh manusia dalam memproses atau memasukkan data secara otomatis dalam sistem komputer.

RPA dapat melakukan proses yang dilakukan secara repetitif seperti memasukkan data, memvalidasi data, mengirim dokumen, dan tugas repetitif yang lain. Dengan menggunakan *RPA*, maka proses rekapitulasi *invoice* dapat dilakukan secara repetitif secara terus menerus dan dapat mengurangi penggunaan sumber daya manusia, sehingga resiko terjadi kesalahan yang diakibatkan oleh manusia akan menjadi berkurang. Kemudian dengan menggunakan *RPA* maka waktu proses rekapitulasi *invoice* yang dibutuhkan menjadi lebih sedikit, sehingga proses bisnis akan semakin lebih ringkas dan secara otomatis akan mengirimkan hasil laporan rekapitulasi *invoice* kepada bagian *finance*.

Pada penelitian ini akan meriset dan mengembangkan sebuah aplikasi menggunakan teknologi *robotic processing automation* pada salah satu proses bisnis yaitu proses rekapitulasi *invoice* dan melakukan validasi serta mengirim informasi hasil rekapitulasi terhadap *invoice* yang telah direkap. Metode penelitian yang akan digunakan adalah riset dan pengembangan dengan menggunakan metode *software development life cycle waterfall* serta akan menguji tingkat efektivitas dari penggunaan *robotic processing automation* dalam proses rekapitulasi *invoice* menggunakan gabungan metode pengujian *white-box* dan *black-box* dengan berfokus kepada hasil penyerapan informasi pada dokumen *invoice* dan hasil dari rekapitulasi yang dilakukan oleh *RPA*.

2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah

1. Bagaimana merancang dan mengembangkan aplikasi *Robotic Process Automation* dalam proses rekapitulasi *invoice*
2. Bagaimana efektivitas penggunaan teknologi *robotic process automation* dalam proses rekapitulasi *invoice*

3 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. *Robotic Processing Automation* menggunakan aplikasi *UIPath Studio Pro*.
2. Jenis *invoice* yang digunakan adalah *freight forwarder invoice*.
3. Proses automisasi menggunakan robot yang melakukan penyerapan informasi menggunakan *optical character recognition* dan *regular expression*.

4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Menghasilkan produk aplikasi menggunakan teknologi *robotic process automation*.
2. Mendapatkan informasi tentang tingkat efektivitas dan kegunaan dari penggunaan *robotic process automation* dalam proses rekapitulasi *invoice*.

5 Manfaat

Manfaat dari penelitian adalah sebagai berikut :

1. Menambah pengetahuan dan informasi tentang teknologi automisasi.
2. Mengembangkan aplikasi dengan menggunakan teknologi *robotic process automation*.
3. Mengetahui tentang efektivitas dari penggunaan teknologi *robotic process automation* dalam melakukan proses rekapitulasi *invoice*.

6 Landasan Teori

6.1 *Robotic Process Automation (RPA)*

Robotic process automation adalah salah satu perangkat lunak otomisasi yang dapat meniru tindakan manusia dengan melibatkan perangkat lunak dalam melakukan interaksi dengan aplikasi komputer dan menyelesaikan tugas yang berbasis aturan seperti membaca, mengetik,

atau mengklik aplikasi yang sudah biasa digunakan sesuai dengan tugas yang diberikan. Beberapa kasus, *RPA* juga dapat melakukan kegiatan perhitungan dan pengambilan keputusan yang kompleks sesuai dengan data dan aturan yang telah ditentukan sebelumnya. Kemajuan teknologi di bidang kecerdasan membuat beberapa aktivitas seperti pemicu, kondisi, dan tindakan dapat dilakukan dengan menentukan pemicu sesuai dengan kondisi dari suatu kegiatan ketika akan dieksekusi, sehingga *RPA* dapat menyelesaikan tugas yang sebelumnya tidak mungkin dilakukan.

Beberapa teknologi yang diadopsi dengan *RPA* seperti *machine learning*, *natural language processing*, *natural language generation*, dan, sehingga terkadang disebut dengan otomisasi cerdas. Dengan munculnya *RPA*, maka proses otomisasi akan menjadi lebih mudah, hanya perlu mengetahui langkah – langkah yang harus diambil oleh manusia dan membuat robot dapat meniru tindakan manusia pada komputer, tetapi dalam beberapa kasus terdapat proses yang memiliki ketentuan seperti prosedur, logika bisnis, validasi, transformasi, dan penggunaan data yang sudah dikodekan oleh sistem.

Platform *RPA* memiliki komponen robot yang dapat berinteraksi dengan aplikasi sesuai dengan langkah – langkah yang dilakukan oleh manusia dan membedakan *RPA* dari proses otomisasi tradisional. Robot dapat dilatih menggunakan langkah – langkah yang bersifat ilustratif dengan menggunakan intruksi berdasarkan kode program (Tripathi, 2018).

6.2 *UIPath Studio Pro*



Gambar 1. Logo *UIPath*

UIPath studio pro adalah salah satu perangkat lunak dalam mendesain dan membuat robot process automation yang membantu pengguna yang belum memiliki keterampilan pengkodean untuk membuat dan merancang *RPA* dengan antarmuka secara visual. *UIPath studio pro* menggunakan pemodelan berbasis diagram alur, sehingga proses pembuatan *RPA* akan menjadi lebih cepat dan mudah. Kemudian dalam proses pembuatan *RPA* dapat dilakukan secara kolaboratif dalam alur kerja yang sama dan dengan tampilan secara visual, dapat menunjukkan dimana letak kesalahan ketika pembuatan *RPA* menjadi lebih mudah. *UIPath Studio Pro* memiliki beberapa komponen pendukung seperti *UiPath Robot* dan *UiPath Orchestrator* (Tripathi, 2018) .

6.3 Optical Character Recognition

Optical character recognition adalah teknologi pada sistem komputer untuk memiliki kemampuan dalam mengekstrak dan memproses teks dari gambar secara otomatis, sehingga data yang didapatkan dari dokumen cetak atau tulisan tangan bisa digunakan kembali sebagai informasi melalui komputer. Salah satu cara sederhana untuk mendapatkan dan menyimpan informasi ke sistem komputer dari dari dokumen berbentuk kertas adalah dengan memindai dokumen dan kemudian menyimpannya sebagai file gambar, akan tetapi untuk mendapatkan informasi tersebut sangat sulit, karena harus mengenali karakteristik font dari setiap karakter pada saat membaca teks atau informasi lainnya dari file gambar.

Kemudian teknologi OCR sangat bergantung pada kualitas dari dokumen kertas dan kualitas gambar, karena terkadang beberapa karakter mungkin tidak dapat dikenali dengan benar oleh sistem komputer, sehingga membutuhkan mekanisme pengenalan karakter, salah satunya adalah dengan mekanisme *Document Image Analysis (DIA)* yang dapat mengatasi masalah tersebut dan menghasilkan format elektronik dari transformasi dokumen dalam format kertas seperti teks tulisan tangan, teks

cetak atau pindaian gambar, ke dalam format digital yang dapat diedit untuk lebih dalam dan lebih jauh untuk diolah. Teknologi pengenalan karakter optik dapat memungkinkan mesin secara otomatis dapat mengenali teks pada beberapa jenis dokumen seperti gambar atau PDF. Kinerja dan akurasi dari teknologi *OCR* secara langsung tergantung sangat bergantung kepada hasil kualitas dokumen masukan. Terdapat beberapa jenis perangkat lunak *OCR* yaitu *Desktop OCR*, *Server OCR*, *OCR web*, dan sebagainya (Hamad & Kaya, 2016).

6.4 Regular Expression (Regex)

Regular expression adalah teknik untuk menemukan teks dengan cara menentukan jenis pola teks yang spesifik dan dapat digunakan pada berbagai bahasa pemrograman. *Regular expression* dapat digunakan untuk memverifikasi apakah teks yang terdapat didalam sebuah informasi cocok dengan pola teks, dapat digunakan untuk mengganti teks yang cocok dengan pola dengan yang lain teks atau bit yang disusun ulang dari teks yang cocok, dan digunakan untuk membagi blok teks menjadi daftar subteks. Dengan menggunakan *regular expression* maka dapat menyederhanakan kode pemrograman dan memproses teks pada suatu informasi, serta dapat mengizinkan banyak tugas yang sama sekali tidak mungkin dilakukan tanpa aturan regular ekspresi.

Tanpa *regular expression*, membutuhkan ratusan baris kode prosedural dalam ekstrak informasi dari dokumen sehingga program akan menjadi membosankan untuk ditulis dan sulit dipertahankan. Tetapi dengan *regular expression* yang tepat, maka hanya membutuhkan beberapa baris kode, atau bahkan mungkin satu baris, tetapi jika hanya mencoba melakukan terlalu banyak dengan hanya satu ekspresi reguler, atau menggunakan ekspresi reguler di teks tertentu. Dengan *regular expression* dalam memanipulasi atau mengekstrak teks di komputer, pemahaman yang kuat tentang regular ekspresi akan menghemat waktu

dan lebih efisien dalam menyerap informasi yang dibutuhkan (Goyvaerts & Levithan, 2012).

6.5 *Software Development Life Cycle Waterfall*

Software Development Life Cycle Waterfall adalah proses pengemabangan perangkat lunak secara sekuensial sesuai dengan kemajuan yang akan mengalir kebawah seperti air terjun melalui beberapa tahapan yang harus dijalankan agar dapat membangun perangkat lunak komputer. Waterfall mendefinisikan beberapa tahapan secara berurutan yang harus diselesaikan tahapan demi tahapan dan pindah ke fase berikutnya jika fase tersebut telah selesai dilakukan, sehingga model ini bersifat rekursif dengan alasan setiap fase dapat diulang tanpa berhenti hingga selesai (Andrei, Casu-pop, Gheorghe, & Bolangiu, 2019).

Model *waterfall* secara dasar memiliki lima fase atau tahapan yaitu analisis kebutuhan sistem, desain sistem, implementasi, pengujian, dan pemeliharaan. Berikut ini adalah penjelasan dari fase dasar pada model *waterfall*:

6.5.1 Analisis Kebutuhan

Analisis adalah tahapan atau fase pertama dalam model waterfall, dengan menentukan persyaratan kerangka kerja dan mengumpulkan data berkaitan dengan kebutuhan sistem yang akan dibuat dan mendokumentasikan hasil dari analisis.

6.5.2 Desain Sistem

Desain sistem adalah tahapan atau fase kedua setelah melakukan dan mendokumentasikan hasil analisis dengan membuat rincian kebutuhan dan membuat kerangka serta menentukan arsitektur yang dapat membantu dalam proses pembuatan perangkat lunak.

6.5.3 Implementasi

Impelementasi adalah tahapan atau fase ketiga dengan melakukan proses pembuatan perangkat lunak berdasarkan desain

sistem yang telah ditentukan pada saat fase sistem desain. Implementasi dapat dilakukan dengan membagi proses pembuatan dalam proyek – proyek kecil atau unit dan akan digabungkan dalam tahap berikutnya.

6.5.4 Pengujian

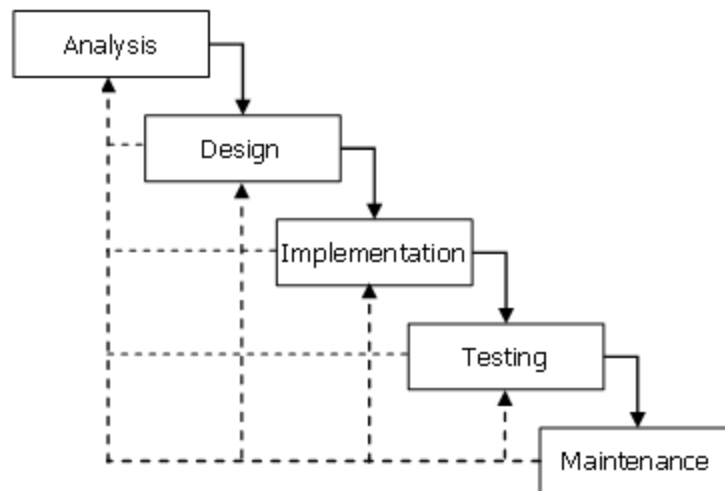
Pengujian adalah tahapan atau fase keempat setelah fase implementasi dilakukan dengan menggabungkan beberapa unit menjadi satu dan dilakukan proses eksekusi untuk mengetahui gabungan dari unit tersebut dapat berjalan, mengalami masalah, atau tidak dapat berjalan.

6.5.5 Pemeliharaan

Pemeliharaan adalah tahapan atau fase kelima untuk memodifikasi perangkat lunak setelah dilakukan perilsan. Pemeliharaan dilakukan untuk memperbaiki kesalahan, masalah, dan meningkatkan kualitas perangkat lunak. Pemeliharaan tambahan dilakukan untuk menyesuaikan perangkat lunak dengan lingkungan sistem dan mengakomodasi kebutuhan dari pengguna baru.

7. Metode Perancangan

Metode perancangan yang akan digunakan adalah menggunakan metode *research and development* dengan jenis metode *software development life cycle waterfall*. Metode *software development life cycle waterfall* adalah metode dengan pendekatan yang sistematis secara berurutan dalam pengembangan perangkat lunak dan sering dinamakan dengan siklus hidup klasik. Metode waterfall bersifat sekuensial yaitu setiap langkah demi langkah harus diselesaikan terlebih dahulu sebelum melanjutkan ke langkah yang selanjutnya dan setiap langkah harus dilakukan secara berurutan.



Gambar 2. Diagram *Software Development Life Cycle Waterfall*

Metode *software development life cycle waterfall* dipilih sebagai metode perancangan dalam penelitian ini karena metode tersebut memiliki alur tahapan pengembangan yang sudah jelas, sehingga proses riset dan pengembangan aplikasi akan menjadi lebih terarah dan dapat didokumentasikan dengan baik. Tahapan pada metode *software development life cycle waterfall* memiliki beberapa tahapan yaitu analisis kebutuhan, desain sistem, implementasi, integrasi atau pengujian, dan pemeliharaan (Andrei, Casu-pop, Gheorghe, & Bolangiu, 2019). Pada tahapan analisis kebutuhan dilakukan untuk mencari dan menentukan kebutuhan dari sistem yang akan diaplikasikan kedalam aplikasi seperti jenis dokumen yang akan dibaca, proses rekapitulasi yang akan

digunakan, dan *bussines exception* yang akan digunakan. Kemudian tahapan desain sistem yaitu mendefinisikan arsitektur yang akan digunakan pada saat proses implementasi, berkaitan dengan algoritma yang digunakan, alur kerja yang dipakai, dan penyesuaian dengan teknologi yang digunakan. Selanjutnya tahapan implementasi yaitu tahapan untuk mengimplementasikan hasil dari desain sistem yang telah ditentukan. Tahapan ini akan melakukan pengkodean dan penyusunan alur proses yang telah ditentukan. Kemudian tahapan pengujian akan dilakukan setelah tahapan implementasi selesai, tahapan pengujian akan menguji aplikasi berkaitan dengan alur proses, fungsionalitas, hasil pembacaan dokumen, dan hasil rekapitulasi. Terakhir adalah tahapan pemeliharaan yaitu tahapan pengembangan lanjutan setelah produk selesai dikembangkan dan dirilis, sehingga aplikasi akan tetap kuat dan dapat berjalan terus menerus.

Pada tahapan pengujian, tidak hanya menguji aplikasi saja, tetapi juga akan melakukan penelitian untuk mengetahui bagaimana tingkat efektivitas dari penggunaan teknologi *robotic process automation* dalam proses rekapitulasi *invoice* menggunakan gabungan metode pengujian *white-box* untuk melakukan pengecekan terhadap kode dan alur program, sedangkan pengujian *black-box* untuk melakukan analisis serta pengecekan terhadap hasil dari kode dan alur program. Pengujian tersebut akan menghasilkan data analisis berdasarkan hasil dari penyerapan informasi dokumen *invoice* dan hasil rekapitulasi dokumen *invoice* yang telah direkap. Sehingga dengan melakukan pengujian dan hasil data tersebut dapat dilakukan analisis terhadap tingkat efektivitas dari penggunaan teknologi *robotic process automation* dalam proses rekapitulasi *invoice*.

8. Teknik Rencana Pelaksanaan

Teknik rencana pelaksanaan pada penelitian ini terdapat pada Tabel 1. Rencana Pelaksanaan di bawah ini :

Tabel 1. Rencana Pelaksanaan

Tahapan Kegiatan	Tanggal Pelaksanaan															
	September 2020				Oktober 2020				November 2020				Desember 2020			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																
Penentuan judul																
Penulisan proposal																
Pengumpulan proposal																
Analisis dan perancangan																
Desain sistem																
Implementasi																

Tahapan Kegiatan	Tanggal Pelaksanaan															
	Januari 2021				Februari 2021				Maret 2021				April 2021			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Implementasi																
Pengujian																
Perilisan																

9. Daftar Pustaka

- Andrei, B. C., Casu-pop, A. C., Gheorghe, S. C., & Bolangiu, C. A. (2019). A Study On Using Waterfall and Agile Methods in Software Project Management. *Journal of Information System & Operations Management*, 125 - 135.
- Goyvaerts, J., & Levithan, S. (2012). *Regular Expressions Cookbook, Second Edition*. Sebastopol: O'Reilly Media, Inc.
- Hamad, K. A., & Kaya, M. (2016). A Detailed Analysis of Optical Character Recognition Technology. *International Journal of Applied Mathematics, Electronics and Computers*, 244-249.
- Singh, A., & Kaur, P. J. (2017). A Simulation Model For Incremental Software Development Life Cycle Model. *International Journal of Advanced Research in Computer Science*, 126-132.
- Tripathi, A. M. (2018). *Learning Robotic Process Automation*. Birmingham: Packt Publishing.

A STUDY ON USING WATERFALL AND AGILE METHODS IN SOFTWARE PROJECT MANAGEMENT

Bogdan-Alexandru ANDREI^{1*}

Andrei-Cosmin CASU-POP²

Sorin-Catalin GHEORGHE³

Costin-Anton BOLANGIU⁴

ABSTRACT

There are many project management methodologies which one can choose from when starting a new project. The purpose of this article is to analyze the most known methodologies, Agile and Waterfall, in order to determine which is the most suitable for a software project.

We propose a practical study by analyzing the results of a survey designed to capture the experience of developers with the aforementioned methodologies. We will focus on the Scrum and Kanban Agile methods and Waterfall to analyze the findings of the study.

Given the results of the study, we concluded that there is no silver bullet solution when it comes to choosing the methodology for a project, as numerous factors need to be accounted for. Waterfall will be a better solution for small projects that have well-defined requirements that will not change, while Agile is preferred when continuous delivery and feedback are important, requirements are not well defined and time to market is more important than releasing a full feature version.

KEYWORDS: *Agile, Scrum, Kanban, Waterfall, Project Management*

INTRODUCTION

Kanban [1] was first introduced as a scheduling method for assembly lines in Toyota factories. It was developed to improve the workflow and to maintain a high level of production. Despite its modest roots, it was quickly acknowledged worldwide as an efficient way to “organize” projects. Kanban was transformed into an abstract concept that could be applied to different sectors and industries, thanks to its efficiency.

^{1*} corresponding author, Student, “Politehnica” University of Bucharest, 060042 Bucharest, Romania, alex.bogdan.acs@gmail.com

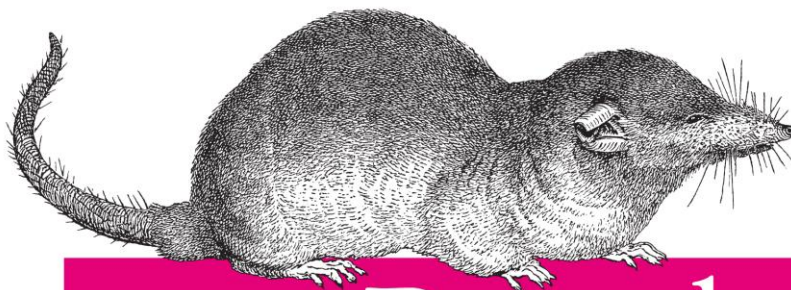
² Student, “Politehnica” University of Bucharest, 060042 Bucharest, Romania, casupopandrei@gmail.com

³ Student, “Politehnica” University of Bucharest, 060042 Bucharest, Romania, gheorghe.catalin97@gmail.com

⁴ Professor PhD Eng., “Politehnica” University of Bucharest, 060042 Bucharest, Romania, costin.boiangiu@cs.pub.ro

*Detailed Solutions in Eight
Programming Languages*

2nd Edition
Revised and Updated



Regular Expressions Cookbook

O'REILLY®

*Jan Goyvaerts
& Steven Levithan*



A Detailed Analysis of Optical Character Recognition Technology

Karez Abdulwahhab Hamad ^{*1}, Mehmet Kaya ²

Accepted 3rd September 2016

Abstract: In many different fields, there is a high demand for storing information to a computer storage disk from the data available in printed or handwritten documents or images to later re-utilize this information by means of computers. One simple way to store information to a computer system from these printed documents could be first to scan the documents and then store them as image files. But to re-utilize this information, it would be very difficult to read or query text or other information from these image files. Therefore a technique to automatically retrieve and store information, in particular text, from image files is needed. Optical character recognition is an active research area that attempts to develop a computer system with the ability to extract and process text from images automatically. The objective of OCR is to achieve modification or conversion of any form of text or text-containing documents such as handwritten text, printed or scanned text images, into an editable digital format for deeper and further processing. Therefore, OCR enables a machine to automatically recognize text in such documents. Some major challenges need to be recognized and handled in order to achieve a successful automation. The font characteristics of the characters in paper documents and quality of images are only some of the recent challenges. Due to these challenges, characters sometimes may not be recognized correctly by computer system. In this paper we investigate OCR in four different ways. First we give a detailed overview of the challenges that might emerge in OCR stages. Second, we review the general phases of an OCR system such as pre-processing, segmentation, normalization, feature extraction, classification and post-processing. Then, we highlight developments and main applications and uses of OCR, and finally, a brief OCR history are discussed. Therefore, this discussion provides a very comprehensive review of the state-of-the-art of the field.

Keywords: OCR, OCR Challenges, OCR Phases, OCR Applications, OCR History.

1. Introduction

It is natural and accustomed that we should demand to build and design machines that can recognize patterns. From automated optical character recognition to face recognition, fingerprint identification, speech recognition, DNA sequence identification and much more, it is clear that accurate and reliable pattern recognition by machine would be greatly useful.

Optical character recognition is an active research area that attempts to develop a computer system with the ability to extract and process text from images automatically. These days there is a huge demand for storing information to a computer storage disk from the data available in printed or handwritten documents to later re-utilize this information by means of computers. One simple way to store information to computer system from these paper documents could be to first scan the documents and then store them as image files. But to re-utilize this information, it would be very difficult to read or query text or other information from these image files. Therefore a technique to automatically retrieve and store information, in particular text, from image files is needed. Of course, this is not a very trivial task. Some major challenges need to be laid out and handled in order to achieve a successful automation. The font characteristics of the characters in paper documents and quality of images are only some of the recent challenges. Due to these challenges, characters sometimes may not be recognized correctly by computer system. Thus there is a need of mechanisms of character recognition to perform Document Image Analysis (DIA) which overcomes these challenges and produces electronic format from the transformed documents in paper format [2].

Similarly, Optical Character Recognition (OCR) is the process of modification or conversion of any form of text or text-containing

documents such as handwritten text, printed or scanned text images, into an editable digital format for deeper and further processing. Optical character recognition technology enables a machine to automatically recognize text in such documents. In real world example, it is like combination of mind and eye of human body. An eye can detect, view and extract the text from the images but absolutely the human's brain processes that detected or extracted text read by eye [1]. Of course OCR technology has not advanced enough to compete with human's ability. The performance and accuracy of OCR is directly dependent upon the quality of input documents. Again, when we think of human's ability to recognize text, the performance of brain's process directly depends upon the quality of the input read by eye. While designing and implementing a computerized OCR system, several problems and challenges can occur. For example there is very slight difference between some digits and letters for computers to recognize them and distinguish one from the others correctly. For example, it may not be very easy for computers to differentiate between digit "0" and letter "o", especially when these characters are embedded in a very dark and noisy background. One of the main focuses of OCR research has been to recognize cursive scripts and handwritten text for its broad application area. Today, to solve the text recognition problem several different types of OCR software exist such as Desktop OCR, Server OCR, web OCR and so on.

Since the OCR research is an active and important field in general pattern recognition problems, due to its fast progress, comprehensive reviews of the field are needed on a regular basis to keep track of the new advancements. One such review was published to discuss the challenges with text recognition in scene imagery [2]. This paper attempts to elaborate on these kinds of studies by providing a comprehensive literature review of optical character recognition research. We discuss major challenges and main phases of optical character recognition such as pre-processing, segmentation, normalization, feature extraction, classification and post processing in detail which needs to be considered during implementing any application related to the OCR, and in the last section of our paper some OCR applications

¹ Software Engineering Department, Firat University, Elazig, Turkey

* Corresponding Author: Email: karez.abdulwahhab@gmail.com

Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.



A SIMULATION MODEL FOR INCREMENTAL SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

Amminder Singh
Research Scholar
UIET Panjab University
Chandigarh, India

Puneet Jai Kaur
Assistant Professor
UIET Panjab University
Chandigarh, India

Abstract: Software Development Life Cycle, SDLC for short, is a very much characterized, organized sequence of phases in software engineering to add to the proposed software product [12]. SDLC gives a progression of ventures to be taken after to plan and add to a software product effectively. This research deals with a fundamental and significant issue in computer world. It is concerned with the software management processes that study the area of software development through the development models, which are known as software development life cycle (SDLC). This paper also describes the software development life cycle model of software used in the field of software development. These models are categorized into two types: traditional Models and contemporary models. Waterfall model, incremental model, spiral model and the V-shaped model are common under traditional model. The contemporary model consists of agile model and Extreme programming model. These are widely used in industry. In this paper, we are proposing a simulation model for iterative process using .NET simulation tool. The purpose of the research is to determine how to get the maximum productivity by using less number of resources, in minimum time.

Keywords: Software Engineering; SDLC; Incremental model; Computer Simulation; .NET;

I. INTRODUCTION

The process of setting up a computer software and information systems is determined by the different ways of development methodologies. To plan, manage, and control the process of developing information system software development methodology framework has been used. A software development technique, officially known as SDLC, mainly used in systems engineering, software engineering, mechanical engineering, computer science, computational science and engineering applications, and some engineering and industrial fields [1]. SDLC is a structure based on development of a software product. It is a group of system development life cycle (SDLC) [2]. As the need of computers have increased day by day. Thus, Computers find their applications in different areas like in banking, medical field, educational field etc. All these areas require software according to their applications and need. Hardware alone is not fulfilling all the needs to do some useful work. So, Software must be used in addition with the hardware. To provide quality of product an engineering field known as software engineering is used. Software engineer will provide the quality of product, reliability of the product and estimated budget. To provide all these needs every company has to go through different process. Therefore, software model is used to provide a method for developing a software product.

SDLC is essential because it tells the software developers that what they exactly need from software development program. SDLC has broken the whole cycle of software development and thus the SD (Software development) has been evaluated properly and helps the programmers to evaluate the program properly. SDLC also provides a spare time when the software will be available for use. SDLC works in steps. So if any of the steps get missed the whole outcome gets affected.

The aim of the research work is to identify the optimal number of resources in minimum time. The size of the company varies according to the number of employees that work on various projects (Small, medium and large). We are using the resources effectively so that quality of work and productivity of the company rises up.

This paper is organized as follows: The brief review of SDLC was given in section 1. The description about SDLC model and related work was given in section 2 and section 3 respectively. Section 4 presents the simulation model and experimental results. Finally, the conclusion is made based on the results and discussion in section 5.

II. SDLC MODELS

Software development life cycle (SDLC) is a process used by many industries to design, develop and test the quality of the software product. The main aim of the SDLC is to provide a high quality product. According to IEEE standard glossary of software engineering terminology, the software life cycle is defined as the time between when a software product is conceived and when the product is no longer available for use. 'Software life cycle typically includes the following activities:

- Requirements Analysis
- Specification
- Software architecture
- Implementation
- Testing
- Documentation
- Training and Support
- Maintenance

The requirement gathering and analysis phase helps to understand the problem. A plan has been made to solve this

Alok Mani Tripathi

Learning Robotic Process Automation

Create Software robots and automate business
processes with the leading RPA tool - UiPath



Packt>