ISSN : 2350-0743

www.ijramr.com

International Journal of Recent Advances in Multidisciplinary Research
Vol. 04, Issue 12, pp.3215-3219, December, 2017

RESEARCH ARTICLE

SPARE PARTS DEMAND FORECASTING SYSTEM

*,[1]Bethala Shirisha and [2]Bethala Pravallika

[1]Assistant Professor, MTech, Department of CSE, CMRIT
[2]Assistant Professor, MTech, Department of IT, IARE

ARTICLE INFO

Article History:
Received 10th September, 2017
Received in revised form
13th October, 2017
Accepted 18th November, 2017
Published online 30th December, 2017

Keywords:

Dashboard, Forecasting,
Spare Parts, SDLC.

ABSTRACT

Reordering spare parts for the purposes of stock replenishment is an important function of the parts manager in the typical dealership. Meaningful reordering requires a reliable forecast of the future demand for items. A Variety of alternative forecasting techniques were evaluated for this purpose with the aim of selecting one optimal technique to be implemented in an automatic reorder in module of a real time computerized inventory management system. Shortage of spares lead to lost sales opportunities and customer dissatisfaction whereas excess inventory meant prohibitive inventory carrying costs. To address this challenge, an effective demand forecasting system is required, which can:

- Provide multiple methods of forecasting
- Provide a part wise dashboard indicating demand trends in graphical format.

INTRODUCTION

Satisfaction of customer in automotive industry depends on the availability of product, which is connected to the quality of product characteristics and product support. Awareness to customer about quality of product is affected by how fine it conforms to requirements that fits to the intended use and also the overtime reliability. Satisfaction of customer is affected by characteristics of products like product support, maintainability, and supportability. In addition, it not only affects the value and performance of hardware purchased, but also its total value received, and also the interaction quality and experience throughout the service life of product. Therefore, products in automotives, specially the spare parts need support to work efficiently for lifetime. Various support that manufacturer offers customers that help them gain maximum worth from product are known as product support. Mostly industrial products and automotive products wear and deteriorate with usage of time. Generally due to economic and technical consideration, as it is difficult to design a machine or product that will be maintenance free. An automotive major has a challenge to manage the smooth availability of spare parts across all the regions to ensure high customer satisfaction levels. Shortage of spares lead to lost sales opportunities and customer dissatisfaction whereas excess inventory meant prohibitive inventory carrying costs. To address this challenge, an effective demand forecasting system is required, which can:

- Provide multiple methods of forecasting
- Provide a part wise dashboard indicating demand trends in graphical format

Corresponding author: **Bethala Shirisha**
Assistant Professor, MTech, Department of CSE, CMRIT

Users of the System

Multiple profiles of users shall operate this system.

- Supply Chain Function Users
- Servicing Function Users

System Design

Logical Design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams.

Physical Design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

- Input requirement,
- Output requirements,
- Storage requirements,
- Processing Requirements,
- System control and backup or recovery

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

- User Interface Design
- Data Design
- Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase[1]. Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

Input & Output Representation

Input Design: The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Objectives

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information[2].
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

Process Model used with Justification

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

SDLC (Spiral Model)

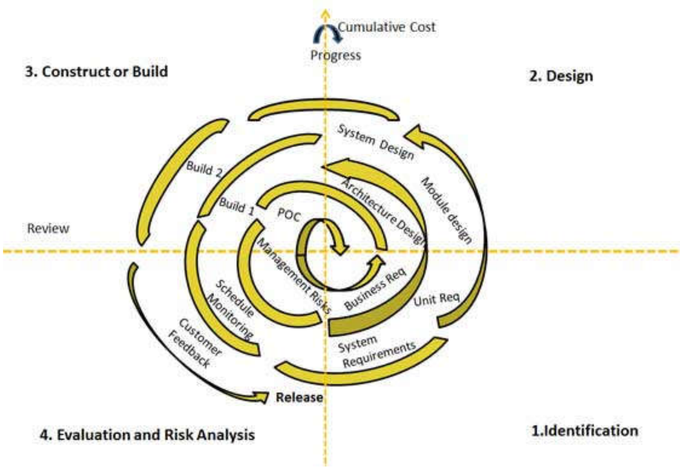

Figure 1. Spiral Model

Stages of SDLC

Requirement Gathering and Analysis

- Designing
- Coding

- Testing
- Deployment

Requirements Definition Stage and Analysis

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

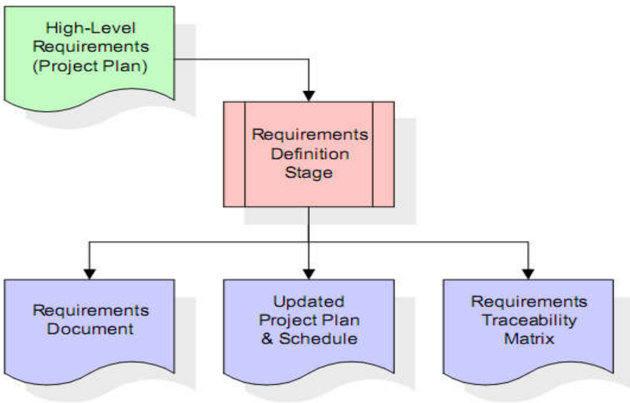

Figure 2. Requirement Definition And Analysis

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). the requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages. In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*. The outputs of the requirements definition stage include the requirements document,[3] the RTM, and an updated project plan.

Design Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary[4]. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

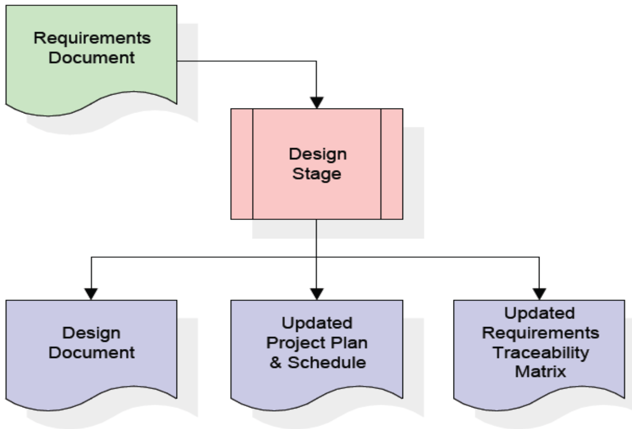

Figure 3. Design Stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions.[5] Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

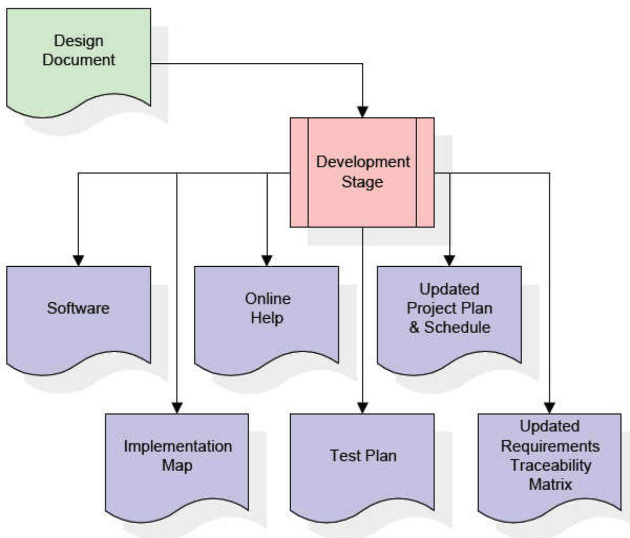

Figure 4. Development Stage

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions,[6] a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan [7].

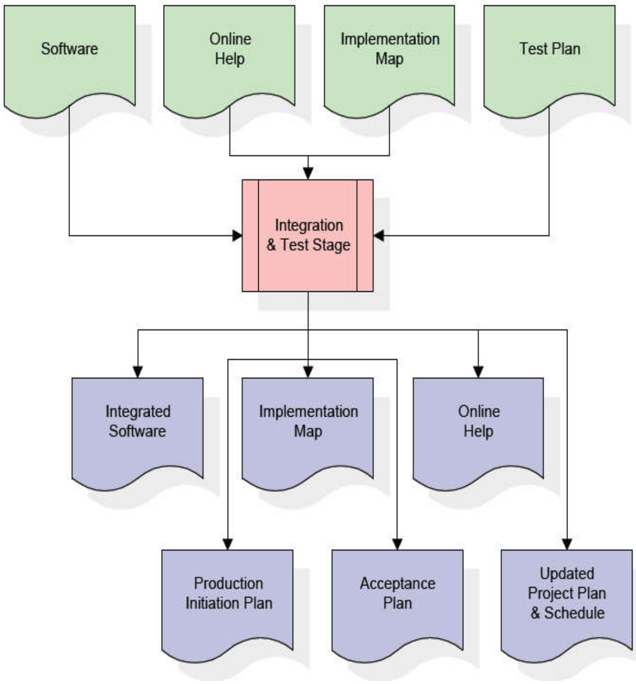

Figure 5. Integration & Test stage

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Stage

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer. After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

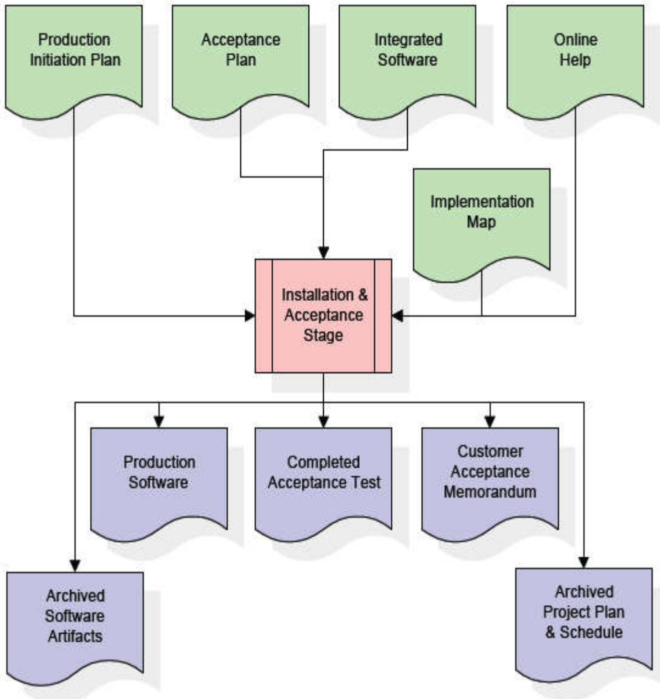

Figure 6. Installation & Acceptance Stage

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software[8]. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

System Architecture

Architecture Flow

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer, data link layer. This project was developed using 3-tier architecture.

Tier Architecture

The three-tier software architecture (a three layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.
The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to

the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems.

Advantages of Three-Tier

- Separates functionality from presentation.
- Clear separation - better understanding.
- Changes limited to well define components.
- Can be running on WWW.
- Effective network performance**.**

Conclusion

The Project objective is to manage the smooth availability of spare parts across all the regions to ensure high customer satisfaction levels. Shortage of spares lead to lost sales opportunities and customer dissatisfaction whereas excess inventory meant prohibitive inventory carrying costs.

Limitations

- Requires a server that needs to be online every time.
- Unpredictable demand of spare parts requires often the transport.

REFERENCES

[1] Morris, M. 2013. Forecasting Challenges of the Spare Parts Industry. *Journal of Business Forecasting*, 32, 22-27.

[2] *International Journal of Production Economics*, 76, 201-215. http://dx.doi.org/10.1016/S0925-5273(01)00174-8.

[3] Willemain, T.R., Smart, C.N. and Schwarz, H.F. 2004. A New Approach to Forecasting Intermittent Demand for Service Parts Inventories. *International Journal of Forecasting*, 20, 375-387.

[4] Kennedy, W.J., Patterson, J.W. and Fredendall, L.D. 2002. An Overview of Recent Literature on Spare Parts Inventories.

[5] Porras, E. and Dekker, R. 2008. An Inventory Control System for Spare Parts at a Refinery: An Empirical Comparison of Different Re-Order Point Methods. *European Journal of Operational Research*, 184, 101-132.

[6] Şahin, M., Kızılaslan, R. and Demirel, Ö.F. 2013. Forecasting Aviation Spare Parts Demand Using Croston Based Methods and Artificial Neural Networks. *Journal of Economic & Social Research*, 15, 1-21.

[7] Kontrec, N.Z., Milovanović, G.V., Panić, S.R. and Milošević, H. 2015. A Reliability-Based Approach to Nonrepairable Spare Part Forecasting in Aircraft Maintenance System. *Mathematical Problems in Engineering*, 2015, 1-7. http://dx.doi.org/10.1155/2015/731437 Nuqul Tissues. http://www.nuqultissue. com

[8] Krajewski, L.J., Ritzman, L.P. and Malhotra, M.K. 2012. Operations Management: Processes and Supply Chain. 10th Edition, Prentice Hall, New Jersey.

****