Abstract- Enterprise applications are complex systems that are hard to develop and deploy in organizations. Although software application development tools, frameworks, methodologies and patterns are rapidly developing; many projects fail by causing big costs. There are challenging aspects that programmers and designers face while working on enterprise applications. In this paper, we present the three of the significant issues: Structural, technically and performance. The important subjects in each aspects are pointed out and recommendations are given. In Structural issues the lifecycle, meta-architecture, guidelines are pointed out. .NET and Java EE platforms are presented in technological issues. The importance of performance, measuring performance and profilers are explained in performance aspects.

Index Terms- enterprise applications, structural, technically, performance

I. INTRODUCTION

Enterprise applications identify the main components of organizations, information systems and how the components including staff, technology, business and resources work together to achieve business objectives. Enterprise applications are very complex systems that are hard to design and implement. Software development and software architecture have received much attention in the last decade even in highly respected big companies and small software firms in all over the world. The growing role of designing and organizing the system before coding is definitely covered and the importance of software architectures, design principles, design patterns etc. is understood exactly. Many design tools, frameworks, design patterns are being developed for designing software systems but unfortunately still lots of projects fail because of many causes. The Chaos Report in 2004 states that the project success rate is 34 percent of all projects. The project failure rate is 15 percent of all projects. 51 percent of all projects are over time, over budget or lacking critical features and requirements. According to the success and failure percentages, it can be indicated that it is very hard to achieve success in enterprise applications. A large number of people at different backgrounds are involved in enterprise applications. There are complex businesses, management and technical aspects which are difficult to control. Also it takes 6-7 years to complete an enterprise application from early design to successful company transformation. Therefore, it is not unusual to have so many problems like over-budgeting, over-time in enterprise applications. There are many pitfalls, bottlenecks and confusing works from beginning to end of an enterprise application. In this paper we propose to analyze the significant issues in enterprise applications. In This paper is structured as follows: In Section 2, the Structural aspects in enterprise applications are presented. In Section 3, the technically aspects including Java EE and .NET platforms are pointed out and the frameworks in Java EE are mentioned. In Section 4, the performance aspects are presented.

II. STRUCTURAL ASPECTS

Structural Aspects are defined in the phases of software lifecycles. The software lifecycle is an abstract representation of software process that defines the software development strategy, steps, methods, activities and product of a software application. Developing an enterprise application begins with analyzing and organizing the elements according to the requirements and sources. Then, comes, designing the system, defining the true architecture and implementing the architecture in high quality. The traditional lifecycle phases of a software application are shown in Fig. 1. The first step of a good architecture is well done project Planning. In the beginning phases, organizational units and business functions to be supported by the system are defined. Also the technical environment and draft project plans are described. In requirement analysis phase, lack of understanding and communication with customer is an important mistakes that affects the architecture or structural of enterprise applications. It is impossible to think through all the aspects that users need properly; but it is important to understand the requirements correctly. The necessities of users change continuously and this causes redesigning the architecture. Designing the system according to the wrong/changing requirements causes headaches in most of the enterprise applications. If a project deviates too far away from original specifications and does not meet the user requirements, it fails because of being late or over-budget. The solution is communicating the users more often and to get requirements correctly. Making the right architectural decisions is very important. Today’s software can be legacy system of tomorrow; therefore good architectures are needed. When beginning a project, the decision of using an existing structural/framework or designing a new one must be made. Architectural decisions are made according to the requirements at different levels. Since architecture is the structural elements of the system together with their externally visible properties and relationships, high level and low level decisions must be made. High level decisions are related with the integrity and structure of the system which is called “meta-architecture”. Meta-architecture involves style,
Software architecture should be designed well and it should be supported with design patterns, reusable class libraries that allow great flexibility for the project. Before the enterprise software architecture is designed, some of the existing architectural frameworks such as MVC, PCMEF and XWA should be analyzed. MVC is Model–View–Controller paradigm that separates View from Model. Model is the non-visual object that consists of application data. View is responsible for showing Model data in a user interface. Taking input from user, managing the model and updating View is a Controller’s responsibility. PCMEF is layered paradigm that consists of presentation, control, domain, domain and foundation layers. The aim of PCMEF is minimizing package coupling, decreasing dependency and increasing stability with using ownward dependencies (higher layers depends on lower layers). When upper layers are changed, lower layers are not affected; this provides loose coupling and allows programmer to build roundtrip architectural modeling. XWA (Extensible Web Architecture) is based on MVC and PCMEF combines the advantages of these frameworks.

III. TECHNICALLY ASPECTS

Enterprise applications are developed with development platforms. A development platform includes programming language(s), run-time environment, and reusable class libraries. There are two leading technologies in enterprise-level application development: .NET and Java EE. In addition, there are alternative technologies like WebObjects, Coldfusion, and PHP (Hypertext Preprocessor). .NET is a Microsoft product described as the software that connects information, people, systems and devices. Java EE is a set of specifications for developing enterprise-level applications, created by the Java Community Process (JCP). We will analyze .NET and Java EE platforms according to dependency, vendor, usage of web services, cost and security.

.NET is a language independent platform that allows programmers to use different properties of programming languages such as C#, VB.NET # (Java for .Net) etc. Besides, .NET is tied closely to the Windows operating system. Microsoft .Net provides good solutions for enterprise applications (with limited choice and limited influence on future directions but the benefits of one source and a known supplier). Java EE is platform independent that runs on any operating system. However, only java can be used as a programming language. This property of Java EE provides an advantage in heterogeneous environments that include different platforms. Also one of the main advantages of Microsoft.NET is its integrated support for web services. Java Platform achieves this with many components. Since Java has the disadvantage of being developed long before Web Service Standards are set, there is not an integrated architecture for web services in Java EE. However Sun has taken aggressive steps to incorporate Web Services into the Java EE standard. .NET utilized the past experience of Java EE, so while java evolves its security capabilities gradually, .NET incorporated more security capability into its original design. Because of simpler and clearer design, .NET provides advantages in security and scalability. Another important point is the Internet Information Server (IIS) that is the web server of .NET. It is one of the most attacked server software in the world.

Java EE and .NET Enterprise applications are generally considered as multi-tiered applications that consist of three tiers: application (client) tier, middle tier and EIS (Enterprise Information System) tier. The tiers in Java EE and .NET are
The application tier is the client side of application. In Java EE, client tier includes java applets, web browsers and java applications. In .NET, there are applications developed with Windows forms and web browsers in client tier. The second tier is the middle tier, which is divided into two parts: web tier and business tier. In web tier, Java EE uses JSP and Servlets, but .NET technology offers ASP.NET. Business tier performs data logic and business processing which are the core functionalities of an application. In Java EE, business code is handled with Enterprise Java Beans (EJB). .Net offers .NET Managed Components as a business tier component of enterprise architecture. The third layer is EIS tier, which consists of database servers, enterprise resource planning systems, and other legacy data sources, like mainframes. Most of the large corporations have existing codes. Both of the technologies offer solutions for legacy integration. .NET uses Host Information Server to connect to legacy systems; but Java EE offers Java EE Connector Architecture (JCA) for integration.

Eat and wheat flours are the integral of daily diet of Indian population. The wheat kernels are processed in chakki (flour mill) to produce wheat flour which is then used to make breads, biscuits, pastas etc. In India chapatti and other variants of wheat forms the staple food of majority of population. In manual process the flour is produced by hand cranking the conventional stone wheels. But this method is characterized by slow operation, low production rate. Further this hand cranking process is physically demanding through energy and postural requirements. It may also leads to clinical and anatomical disorders which may affect operator’s health.

In order to make it possible to operate the system effectively and efficiently it is necessary to develop this system by giving due respect to human limitation. Hence ergonomic system of pedal operated flour mill is developed. The ergonomic consideration mainly includes the selection of components of system which suits the human capability and develops the posture to operate system to reduce the fatigue and chances of muscletolo disorders.

### IV. FRAMEWORKS IN JAVA EE

In Java EE there are many third-party products and open source frameworks. In this section, we will briefly present the third-party products and frameworks used with Java EE. There are different technological choices in Java EE becoming popular in recent years. The open source frameworks and third party products in each tier are shown in Fig. 4.

![Fig. 3 Java EE and .NET tiers](image)

### V. PERFORMANCE ASPECTS

It is important to make the performance tuning in early cycles of the project. If there are architecture mistakes or dangerous bottlenecks, it is beneficial to catch them in early cycles of the project and redesign the architecture. For getting real evidence and ensure level of performance, programmers should use some benchmarks such as web-load testing tools (Microsoft Web Application Stress Tool (WAS), Apache JMeter, etc...). These tools create multiple connections to the web application like real applications in production. Purpose of this test is to measure or observe behaviors of a web. Profiling can indicate which pieces of slow code matter.

### VI. CONCLUSION AND FUTURE WORKS

The architecture of enterprise applications is divided into three tiers which are application (client) tier, middle tier and EIS (Enterprise Information System) tier. In Java EE, there are many third-party products and open source frameworks for each tier. This provides alternatives to programmers while implementing an application. Performance determines the success or failure of a project. First, efficient and suitable software architecture should be chosen. If architecture is designed well and it is supported with design patterns, reusable class libraries allow great flexibility for the project. If a project does not meet performance requirements, the whole project goes for nothing and it fails. Determining the performance is not feasible by just looking at the code; to have real evidence about performance the code should be run to get the right performance. As future work, we will implement an enterprise web application by following the architectural, technological and performance issues presented in this paper. We propose to use open source third party Java EE frameworks (Spring, JSF and Hibernate) and implement a enterprise application to be used in municipalities.

### REFERENCES


AUTHORS

Tejinder Singh , Research Scholar of Computer Science, JJTU, Jhunjhunu Rajasthan, INDIA
Email id - tejinder31.singh@gmail.com