

Managing Software Products and Product Lines of Web-Based and Cloud-Based Applications

Final Term Paper - Spring 2023

Software Project Management - CS 587

Illinois Institute of Technology, Chicago

Team 7:

Avula, Prabhu - A20522815

Bolla, Nagarjuna - A20524548

Parthasarathy, Punalraj - A20519421

Introduction

Businesses worldwide are rapidly moving to the Internet to conduct their business. We have seen unprecedented growth in the past few decades, which will keep growing. The Internet offers everything at the click of a button. You could be anywhere worldwide and access information, purchase products, or stream entertainment. The driving force, the backbone of the Internet age, is Web-Based Applications. Websites that look alluring to the eye but are complex behind the scenes. Several engineers made efforts to give the user a fruitful and wonderful experience.

In recent times, however, another force has become the present and likely the near future, Cloud-Based Applications. With cloud-based applications, you do not need to worry about your information as they are stored and locked away, safe, in a cloud server, enabling you to access them from anywhere in the world as long as you are connected to the Internet.

Also, businesses have a way of capitalizing on the market's needs. Companies often have a standard, recurring theme in almost all business avenues in their products. An essential, fundamental part that connects each product. This is the product line, the backbone for almost all major businesses. Whether you are writing a paper, preparing a presentation, or crunching numbers on a document, chances are you are using a suite of applications or a Product Line. Other examples include the line of beverages provided by Starbucks, clothing, footwear, and accessories specializing in sports by Nike, Adidas, or Puma.

In this paper, we will discuss Web-Based and Cloud-Based Applications and Product Lines and learn about their relationship with software products, projects, and, most importantly, managing it all. We will make a precise, comparative analysis between the two and understand how all of these things are integrated to guarantee the user a smooth and enjoyable experience.

Discussion of Relevant Topics

Before we discuss the comparative analysis between web and cloud-based applications, we must understand the required terminology and the key points, such as the advantages and differences between these two types of applications. It is crucial to have a basic, foundational sense of what software project management is, what web-based and cloud-based applications are, what product lines are and how they work, and how they impact our daily lives. So, let us get right into these topics.

Software Project Management & Product Lines

Software Project Management, or SPM, is the process of managing a product's life cycle with the help of software. In a realistic scenario, a product cycle has many components. Several employees are working on different aspects of the product. From the person who gathers the business requirements to the person who writes the code to the person who trains the client, many people are involved in a project or a product cycle. Sometimes, using traditional management techniques is not practical. Likewise, while keeping close to traditional textbook management techniques is a good start, the reality is often different. We believe precise judgment and intuition are required to handle real-world scenarios adequately.

Hence, most companies in this day and age make use of software to help the project manager handle everything and everyone. Examples of SPM software would be MS Project (by Microsoft), Confluence, Teamwork, etc. Integrating this software with human resource management software often helps the company keep track of the project.

Now that we have understood software product management let us talk a little about what product lines are and how they impact our daily routines. Essentially, a line of software-intensive products with a common thread or theme running through them is known as a product line. This is because they take economic advantage of commonality, i.e., if there is a common need for a set of products in the market, something that the majority of end users would want and would be willing to pay for; companies use the product line idea to capitalize on that need. Common examples include MS Office, Nike, Starbucks, Coca-Cola, etc.

Since all the products in the line have a vast commonality, the process of planning, development, building, testing, and deploying is the same. Although slight variations exist between each specific product, the core idea remains intact. This process is known as strategic reuse. The company invests in the core idea and core process for one product and then applies the same across the board. The output, of course, is this fantastic line of similar products that satisfy multiple needs of the everyday consumer.

Whether it is evident or not, product lines directly relate to every living person on the planet. Someone, somewhere, is using something related to a product line. That idea has a

dominant influence over us. Now that we have discussed these topics and understand what SPM and Product Lines are, let us dive into domain-based applications.

Domain-Based Applications - Web Based

Web Based Applications are applications that can be accessed by the user primarily over a network. Whether it is the internet or some server, as long as the contents the user is interested in do not exist in the user's computer, that application is considered a web-based application. They often use protocols such as HTTP, HTTPS, etc., and run inside a browser such as Google Chrome, Safari, Mozilla Firefox, Opera, etc.

Now, the tech stack for a web-based application is as follows. They primarily consist of a front end and a back end. The front end is the end the user interacts with, so it must look captivating and aesthetic. While HTML and CSS are the bread and butter of what makes the front end look and function beautifully, various other languages and frameworks are used. These include AngularJS, ReactJS, VueJS, Ruby, .NET, etc. They are a set of predefined tools, and blueprints, that assist a developer in constructing the web application.

Product Line Technologies in Web-Based Applications are as follows. All web-based applications use a similar approach, so designing a product line system is easy and efficient. However, there are a few subtle differences that give us the three leading product line technologies used in web-based applications:

- **Object Technology:** Also known as object-oriented programming, it categorizes any real-world thing as an object. An object has four key characteristics: state, behavior,

identity, and relation. Furthermore, other concepts such as polymorphism, inheritance, and encapsulation are vital to this approach to product line development. Examples include programming languages such as Java, C++, and Python.

- **Aspect-Oriented Technology:** Instead of viewing real-world things as objects, this approach views a group of them as aspects. AOT usually takes a look at crosscutting concerns throughout the product line structure. It gives a predefined blueprint, a framework for developers to build their applications.
- **Model-Driven Development:** This process focuses on creating and exploiting domain models. It has two phases, namely, abstraction and automation. In the abstraction phase, it organizes complex systems, and in the automation phase, it automates long processes increasing efficiency and reducing costs.

Domain-Based Applications - Cloud Based

Cloud-based applications are a subset of web-based applications. While all cloud-based applications are web applications, not all web applications are cloud applications. The main difference lies in the point that, in cloud-based applications, all your information is stored virtually. In this case, your information will be recovered if there are any issues with your physical storage locations. Prime examples of cloud applications include Google Drive, Dropbox, iCloud, etc.

There are four main types of cloud applications. They are:

- **PaaS - Product as a Service:** In PaaS, the company provides infrastructure, solutions, and cloud computing platforms so the developer can build their product. Microsoft Azure is a perfect example.
- **SaaS - Software as a Service:** SaaS is the most popular form of cloud-based application. They provide full-scale applications to be used entirely on the cloud. Everything the user does is in the cloud. Examples would be CRM and ERP software.
- **IaaS - Infrastructure as a Service:** IaaS is a model where companies outsource their cloud-based infrastructure services. AWS is a good example.
- **RaaS - Recovery as a Service:** Recovery as a service(RaaS) is a cloud computing service model that allows an organization to back up its data and IT infrastructure in a third-party cloud computing environment and provide all the DR orchestration.

Now, let us discuss the tech stack in cloud-based applications. The tech stack remains the same since they are a subset of web-based applications. However, the difference is in the way the respective products are delivered. In cloud-based applications, the delivery is based on three models: Private, Public, and Hybrid.

Product line implementations in cloud-based applications have five primary phases. The whole idea of cloud-based models is that everything the user interacts with on a cloud app is a virtual representation of physical things. This idea is known as virtualization. It enables multiple applications to utilize the same resources efficiently. Fundamental infrastructure does exist, in any case. Data centers across several locations exist to help cope with the required demand.

Furthermore, middleware exists. This middleware is a mediatory phase between the infrastructure and delivery. It helps with seamless communication between applications. Next comes management, the process of monitoring the performance and capacity of the cloud servers. Finally, we have automation software that automates simple tasks, thus easing workload, streamlining delivery, and reducing operating costs for a client.

Comparative Analysis in Web-Based Applications

As discussed previously, web-based applications can be segregated into three main categories:

- Object Technology
- Aspect-Oriented Technology
- Model Driven Development

The best way to compare these three approaches is to weigh their pros and cons. Here is the relevant information in a tabular format:

	Pros	Cons	Examples
Object Technology	<ul style="list-style-type: none"> ● Reduced Maintenance ● Real-World Modeling ● Improved Reliability and Flexibility 	<ul style="list-style-type: none"> ● Steep learning curve ● more susceptible to security breaches. ● Difficulty in 	Java, C++, Python

	<ul style="list-style-type: none"> • High Code Reusability 	initializing base classes. <ul style="list-style-type: none"> • Larger and Slow Running Programs. 	
Aspect Oriented Technology	<ul style="list-style-type: none"> • Compliments Object technology • Modularizes cross-cutting-concerns • Improved Maintainability and Understandability. 	<ul style="list-style-type: none"> • Not well documented • Runtime Overheads 	PHP, Spring, JS, .NET
Model Driven Development	<ul style="list-style-type: none"> • Enables scripters to contribute to enterprise development • Increases flexibility 	<ul style="list-style-type: none"> • The cost of failure is high • Not high-level enough to allow powerful abstractions. 	AngularJS, ReactJS

The table above shows that the object technology approach has reduced maintenance. It offers real-world modelling capabilities, improved reliability and flexibility, and, most importantly, high code reusability, which means that employing this approach for a product line would require a substantial amount of work initially. However, once the foundations are in place, it gets easier. Now, the cons of this approach include a steep learning curve. Furthermore,

product lines using this approach are more susceptible to security breaches. Initializing base classes is also challenging; hence, the programs are large and slow. Examples of object technology-based products include Java, C++, and Python. They are known as Object Oriented Programming Languages.

Now, let us discuss aspect-oriented technology. This approach builds upon the existing technology offered by object-oriented concepts. The end goal is to make a blueprint, a framework of sorts. Something that other developers can avail that will aid them in building their products. Hence, the pros include modularising cross-cutting concerns and improving maintainability and understandability. The cons include huge runtime overheads and that there needs to be more documentation. Examples of such products include .NET, PHP, Springboot, JS, etc. These are indeed frameworks. .NET is perhaps the most famous one. Released by Microsoft, it is one of the most widely used frameworks in the world.

Finally, let us discuss Model-Driven Development. As mentioned earlier, this approach is based on two phases: abstraction and automation. So, with the increased flexibility and time, this approach enables scripters to contribute to enterprise development significantly. On the contrary, the cons to this approach are quite risky. The cost of failing is high in most real-world cases. Moreover, since this approach is relatively newer than the previous two, it has yet to evolve to a level where it can process and allow powerful abstractions. Examples include AngularJS and ReactJS.

So, each approach has its set of pros and cons. While some approaches have been heavily used and are reliable, others are relatively new and offer great scope for growth in the future. While deciding which approach to choose, it is vital for the Project Manager to understand the business requirements and judge which of these approaches offers the best way to build, maintain and expand the product.

Comparative Analysis in Cloud-Based Applications

Drawing comparisons between the three different delivery approaches for product lines in cloud-based applications yields a rather straightforward conclusion. These are:

- ❖ Public
- ❖ Private
- ❖ Hybrid

Let's take a look in detail with the help of the table below:

	Pros	Cons
Public Cloud	<ul style="list-style-type: none"> ● Simple to Implement and Use. Minimal Upfront Costs. ● Widespread Accessibility ● Huge Workloads 	<ul style="list-style-type: none"> ● Expensive in the Long Term. ● Susceptible to service outages.

Private Cloud	<ul style="list-style-type: none"> • Complete control. • Minimal long-term costs. 	<ul style="list-style-type: none"> • The upfront capital required is large. • Needs proper infrastructure (Data Centers) • Cannot handle the huge spike in workload
Hybrid Cloud	<ul style="list-style-type: none"> • Cost efficient as it uses both public and private clouds. • Less susceptible to service outages. • Suited for handling spikes in workload. 	<ul style="list-style-type: none"> • Difficult to implement and manage. • Requires some space for infrastructure.

In the public cloud delivery-based approach, the pros are plentiful. They are easy to implement, requiring minimal upfront costs and capital. Moreover, deploying over a widespread area is easy, and most importantly, they are best suited to handle huge workloads effortlessly. However, those pros stated accompany some cons. Since the upfront costs are minimal, they are harder to maintain in the long run because of the excessive workloads. Managing these workloads would require many resources dedicated to servers, etc. These could lead to reliability issues. Furthermore, since the infrastructure is cheap and weak, they are more susceptible to power outages or external, unwanted interference.

Next, the advantages of the private cloud delivery-based approach include having complete control over the cloud. Anything happening on the cloud is overseen and controlled by a single entity. In most real-world cases, this entity is a company or organization. A disadvantage is that building the required infrastructure requires significant upfront capital. Data centers where all the cloud information is stored are expensive to build. However, once built, this model is inexpensive to maintain. It requires minimal costs in the long run. Privatization further ensures a higher sense of security. One last major disadvantage is that this model type cannot handle a massive spike in workload. It is accustomed to a regular work rate and hence would be caught unaware of a vast, unexpected spike in the workload.

Now, let us discuss the hybrid cloud delivery-based approach. As the name suggests, it is the product of public and private cloud models. It incorporates the best of both to give us significant benefits but also brings its fair share of disadvantages. Pros include cost effectiveness, lesser susceptibility to outages, and the capability to handle massive spikes in workloads.

Furthermore, since it uses both public and private approaches, the costs are lower than private or low than public. Hence, it is cost-efficient. However, it takes much work to implement as combining two existing and established cloud models is highly complex. Likewise, it takes work to manage as well. Moreover, drawing inspiration from the private cloud model, this approach also requires some capital to establish the necessary infrastructure, like a data center.

In summary, these three delivery-based models have their fair share of advantages and disadvantages. Companies often discuss which model best suits their and their customers' needs.

While most companies opt for the private cloud model that enables them to have control over everything, a manager must carefully consider each model, considering the business needs of his company before arriving at a decision.

Web-Based Vs. Cloud-Based - A Comparative Analysis

Now that we have conducted the individual comparative analysis' for both web-based and cloud-based applications, let us discuss the big picture. How do they fare against each other? Here's a simple table to help understand:

Pros		Cons	
Web-Based	<ul style="list-style-type: none"> → Easy to maintain → Less expensive to develop → Limited Scalability 	<ul style="list-style-type: none"> → Limited availability → Works only in browsers. 	
Cloud-Based	<ul style="list-style-type: none"> → Very high uptime → Applications on the cloud can run completely or partially without the internet. → High scalability. → Can run on users' computer systems or providers' web 	<ul style="list-style-type: none"> → Initially requires more capital to develop. → Downtime. 	

	servers.	
--	----------	--

When discussing web-based applications, it is essential to understand that they are the original domain-based applications. Any other type of domain-based application is likely based on the same fundamental principles that web-based applications have. With that in mind, the advantages of web-based applications are that they are primarily easy to build and maintain, both in terms of time and financial resources. However, the scalability is also limited. The disadvantages include limited availability and the fact that, as the name suggests, they work only on web browsers such as Safari, Firefox, and Brave. Whatever occurs on these applications is limited to the boundaries of the web browser.

Next, when discussing cloud-based applications, it is vital to remember that they are an extension of web-based applications. Though they incorporate new principles and technologies, the foundational principles are the same as web-based applications. So, the pros include very high up-time. Also, applications on the cloud can run entirely or partially without the Internet. In cases where there are internet outages, there is no data loss as these applications are independent of the Internet. Furthermore, they are highly scalable and can run on any user's computer systems or provider's web servers. The cons are that they initially require significant capital to develop and maintain, and the downtime is unreliable.

In conclusion, while some actual products in the world are primarily based on web applications, there has been a significant shift towards cloud-based applications. The decision ultimately boils down to how your product can best be utilized. As a manager, you are expected

to look at the big picture and consider the present and future scalability opportunities before choosing whether to base your product solely on the web or the cloud.

Conclusion

The future of domain-based applications is undoubtedly towards cloud computing. It is the next big thing in Information Technology services. That being said, web-based applications will still be around. Not only are they the superset of other domain-based applications, but they are also the foundation for cloud-based applications, and all cloud applications will use web services to cater to their needs. In conclusion, web-based and cloud-based applications both have their own set of advantages and disadvantages. However, the fundamental idea is in understanding that once we combine and integrate them perfectly, they will offer a higher sense of luxury for the user.

Bibliography

Gandhi, Revati, Sukhada Pande, and Mayuree Nandi. "Software Product Line Technologies for Web-Based Application." International Journal of Engineering Research & Technology (IJERT), 2020.

<https://www.ijert.org/research/software-product-line-technologies-for-web-based-application-IJERTCONV8IS05047.pdf>.

Gustavo Sousa. A Software Product Lines-Based Approach for the Setup and Adaptation of Multi-Cloud Environments. Software Engineering [cs.SE]. Université de Lille, 2018. English. <NNT : >. <tel-01808779>

"Developing Software Product Lines." n.d. Accessed March 6, 2023. https://resources.sei.cmu.edu/asset_files/EducationalMaterial/2020_011_001_650228.pdf.

Northrop, Linda. n.d. "Software Product Lines Today's Talk." Accessed April 28, 2023. https://resources.sei.cmu.edu/asset_files/Presentation/2005_017_001_23874.pdf.

Ouali, Sami & Kraiem, Naoufel & Ben Ghezala, Henda. (2011). Framework for Evolving Software Product Line. International Journal of Software Engineering & Applications (IJSEA). 2. 10.5121/ijsea.2011.2204.

Shetty, Jyothi, Deepika Dash, Akshay Joish, and Guruprasad C. 2020. Review of *Review Paper on Web Frameworks, Databases and Web Stacks*. International Research Journal of Engineering and Technology (IRJET) 07 (04). <https://www.irjet.net/archives/V7/i4/IRJET-V7I41078.pdf>.

Di Lucca, Giuseppe A., and Anna Rita Fasolino. 2006. "Testing Web-Based Applications: The State of the Art and Future Trends." *Information and Software Technology* 48 (12): 1172–86. <https://doi.org/10.1016/j.infsof.2006.06.006>.

"Cloud-Based vs Web-Based Applications: A Comparison of Features & Key Aspects." 2019. Financesonline.com. May 16, 2019. <https://financesonline.com/cloud-based-web-based-applications-a-comparison-of-features-key-aspects/>.

VMware. 2021. "What Is Cloud Architecture? | VMware Glossary." VMware. December 16, 2021. <https://www.vmware.com/topics/glossary/content/cloud-architecture.html>.