An Evaluation Model for Application Development Frameworks for Web Applications

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By
Changpil Lee
Graduate Program in Computer Science and Engineering

The Ohio State University
2012

Thesis Committee:
Dr. Rajiv Ramnath, Advisor
Dr. Jay Ramanathan
Dr. Tom Bitterman
ABSTRACT

Web frameworks and web-based content management systems (WCMS) are emerging as the growth of the Internet has continued and its role has increased in the daily life of its users. When choosing a web framework or a WCMS on which to build an application, web application developers usually decide on a framework or a WCMS based on their prior web development expertise, the quality of the available documentation about the framework, or the popularity of the web framework, regardless of its ability to meet the non-functional and functional requirements of the application to be built. The problem is that there are many choices with respect to web frameworks and WCMS for a web developer to select from and the goals must be to select the best framework for the job.

When choosing an framework or WCMS, web application developers must consider the overall effect of the design decisions within the framework, the inherent tradeoffs between quality attributes (such as performance and security), and the tradeoffs required to address user, system, and business requirements. However, the process necessary to find the best framework for an application’s requirements is not well described. There is only limited research on the evaluation of web application frameworks; this makes it hard for developers to choose the right web framework or WCMS.

In this thesis, a new cost-estimation model for web frameworks and WCMS is introduced. The costs in the estimation model are categorized as follows: system costs,
design costs, learning costs, and implementation costs. This cost-estimation model is a framework for comparing the relative costs of web frameworks and WCMS. This cost estimation model was applied in two case studies namely the Polymer Properties Predictor (P4) project and the PFast project, both at the Ohio Supercomputing Center. Finally we used the cost-estimation model to analyze the development of a new web project.
Acknowledgments

I would like to express my sincere gratitude to the best advisor and mentor I could have possibly gotten, Dr. Rajiv Ramnath. His knowledge and critical thinking has been very valuable. His encouragement and confidence has made this thesis possible.

I would also like to thank Professor Jayashree Ramanathan and Dr. Thomas Bitterman for their continued support, insights and suggestions that helped shape my research.

I also really appreciate the support I got from my family, especially my wife, and the God who kept encouraging me constantly and kept me on track to complete this successfully.
Vita

August 2004 ......................... B.S. Computer Science,

Purdue University

September 2009 to present ........... Graduate Student, Department of Computer

Science and Engineering, The Ohio State

University.

Field of Study:

Major Field: Computer Science and Engineering
TABLE OF CONTENTS

Abstract.................................................................................................................................ii

Acknowledgements.................................................................................................................iv

Vita...........................................................................................................................................vi

Table of contents....................................................................................................................vii

List of Figures...........................................................................................................................xiii

List of Tables............................................................................................................................ix

1. Introduction..........................................................................................................................1

1.1. Background.........................................................................................................................1

1.2. Motivation & Problem Statement......................................................................................3

1.3. Objectives and research questions....................................................................................5

1.4. Overall research methodology........................................................................................6

1.5. Definitions of terms............................................................................................................7

1.6. Assumptions.......................................................................................................................8

2. Conceptual Layouts and Related Research.......................................................................10

2.1. Conceptual Layouts...........................................................................................................10

2.1.1. Web Framework..........................................................................................................11

2.1.2. CMS.............................................................................................................................13

2.1.3. Functional Requirements..............................................................................................16
2.1.4. Non-functional Requirements .......................................................... 17

2.1.5. Requirements According to Their Application Domains .............. 21

2.2. Previous Framework Comparison Models ........................................ 23

3. A comparison Model for web application frameworks .......................... 26

   3.1. Introduction ....................................................................................... 26

   3.2. Problem statement .......................................................................... 29

   3.3. Assumption ...................................................................................... 30

   3.4. Analysis of Web Framework ........................................................... 31

      3.4.1 System Costs ............................................................................... 31

      3.4.2 Design Costs ............................................................................... 31

      3.4.3 Learning Costs ............................................................................ 35

      3.4.4 Implementation Costs ................................................................. 37

      3.4.5 Testing Costs ............................................................................... 38

   3.5. Methodologies .................................................................................. 38

4. Case Study: P4 Application .................................................................. 41

   4.1. Introduction ...................................................................................... 41

   4.2. Requirements Analysis ................................................................... 42

   4.3. Cost Analysis with Drupal CMS ..................................................... 45

   4.4. Cost Analysis with Pylon ................................................................. 47

   4.5. Assessments and evaluations .......................................................... 49

5. Case Study: PFast Application .............................................................. 53

   5.1. Introduction ...................................................................................... 53
List of Tables

Table 1.1 Web frameworks and CMSs.................................................................4
Table 2.1 Application domain and NFRs .........................................................22
Table 4.1 Cost Analysis developing P4 web application using Drupal...............46
Table 4.2 Cost analysis: developing P4 web application using Pylon...............48
Table 4.3 Summary for P4 Application Development Costs using Drupal.........50
Table 4.4 Summary of P4 Application Development Costs using Pylon ..........51
Table 5.1 Cost Analysis developing PFast web application using Drupal..........57
Table 5.2 Cost Analysis developing PFast web application using Groovy on Grails........59
Table 5.3 Summary for PFast Application Development Costs using Drupal.....60
Table 5.4 Summary for PFast Application Development Costs using Groovy on Grails.61
List of Figures

Figure 2.1 CMS distribution in Top Million Sites ............................................................15
Figure 2.2 Non-functional requirements tree .................................................................21
Figure 3.1 Development process ..................................................................................27
Figure 3.2 Pop-up login Functionality in Python with Pylon ........................................32
Figure 4.1 P4 web application using Pylon ...................................................................41
Figure 4.2 P4 Use case ....................................................................................................42
Figure 4.3 Functional requirement for uploading MOL input file ..................................43
Figure 4.4 Functional requirement for pasting sample mol data ....................................43
Figure 4.5 Previous file storage system shows an input file and an output file ............44
Figure 5.1 P5 web application using Drupal .................................................................53
Figure 5.2 PFast Use case ............................................................................................54
Figure 5.3 Functional requirements for uploading an Excel input file ..........................54
Figure 5.4 Previous file storage system shows an input file and an output file ..........55
CHAPTER 1

Introduction

1.1 Background

Since the World Wide Web was introduced in the early 1990s, its growth and role in daily life have increased considerably. In the early 1990s, the Web was used for sharing data, news, and documentation via the Internet among university-based scientific departments and physics laboratories. By the mid 1990s, web-based commerce (e-commerce) companies such as Amazon.com had emerged, and at the same time email and instant messaging—both driven by user-friendly web interfaces—both became popular. And while the late 90s saw many new dot-com companies create countless concepts and technologies, many of these companies did not survive because they did not have effective business practices. Accordingly, in the past ten years many companies, such as Google, eBay, Amazon, and Facebook, have found success developing stronger business models that helped make the World Wide Web a more compelling experience.

Most early websites used only HTML to exchange information, and while HTML can effectively share static content such as research documentation and pictures, its applicability for more complex business models is limited. Specifically, while early sites used a two-tier (client-server) architecture, in these contexts HTML could not meet "the demand for application integration, flexible architectures, and portable application logic" [1]. Thus, improved three-tier architecture was introduced, consisting of a
presentation layer, one or more dedicated business layers, and a database layer. The presentation layer supports rich user interfaces as well as web application interfaces, while the business logic layer contains generic classes for data retrieval, validation, and posting, enforcing business rules. Finally, the data access layer uses classes, modules, and components to connect to databases. Overall, the three-tier architecture helps business to focus on complex business rules and decouple the database layer and presentation layer from business layer. Apart from the usual advantages of modular software with well-defined interfaces, this architecture is intended to allow any of the three tiers to be upgraded or replaced independently, as requirements or technologies change. For example, a change of operating system from Microsoft Windows to Unix would only affect the user interface code [2].

Many web frameworks emerged because of three-tier architecture, as various companies developed technologies for each specific layer. For instance:

**Presentation layer:** HTML, XML, Silverlight, CSS, JavaScript, Flex, VBscript, Flash, and Applets

**Business layer:** J2EE, Struts 2, Spring, Servlets, JSP, ASP, C#, PHP, and Perl

**Database layer:** mySQL, SQL Server, and Oracle

**Inter communication:** RMI, ESB, AJAX, and XML

For instance, when a web application developer chooses a web application module, he or she can choose to use HTML, XML, J2EE, JSP, Servlets, and mySQL or Silverlight and .NET. In order to develop the web application, he or she can use some of the libraries built in to each software module, but still must build the application from the ground up--
a low-level, tedious, error prone process. Fortunately, many web application frameworks have built-in components (modules) such as a login module, view module, shopping, chat module etc. Thus, much like building contractors can buy (rather than make) components such as windows or flooring, web developers do not need to implement distributed applications from the beginning to achieve their goals. In addition to such frameworks, a content management system (hereafter CMS) makes developing the application systems easier and faster. A CMS is a tool that enables a developer to create, edit, manage and finally publish content such as a text, graphics, video, documents etc [12]. With these tools, a developer can develop web applications much faster and more easily.

1.2. Motivations and Problem

Developers choose a web framework or CMS based on their web developing experiences, the framework’s amount of documentation, or the framework’s popularity (regardless of its functional or non-functional requirements; see section 1.5). Selecting the best of the many frameworks available can pose a problem. For instance, many developers opt for a Linux operating system, using an Apache web server and MySQL database server, but may choose to write in Perl, Python, PHP, Ruby, or J2EE. Or, the developer may choose a Windows OS, using Windows Web Server IIS, a Windows SQL server, and the ASP.net language. Other web server options include Tomcat, Jetty, and Glassfish; alternate database servers include Oracle and DB2. And, as Table 1 shows, each programming language has its own frameworks and CMSs.
Web frameworks and CMSs give web developers many different architectural choices, depending on development purposes and requirements. When choosing an architectural framework, web application developers must consider the overall effect of framework decisions, the inherent tradeoffs between quality attributes (such as performance and security), and the tradeoffs required to address user, system, and business requirements. Before a web architecture designer decides on a framework, several steps must be taken:

- The framework should be studied. There are plenty of open source frameworks. A portfolio for the frameworks should depend on the functional and non-functional requirements.
• The CMS choices and its modules should be searched. If there are some good
CMS choices fulfilling functional and non-functional requirements, it will be it
easier for web developers to implement web applications in the framework.
• A decision on the best framework or CMS should be made for the project.

Unfortunately, this process is not well described, and limited research on the
evaluation of web application frameworks makes it hard for developers to choose the
right web framework or CMS. Previous research conducted by Tuukka Laskso and Joni
Niemi (2008) offers limited comparisons of Java-based frameworks [24], while Lutz
Prechelt and Will Hardy (2007) compared program languages such as Python, PHP, Perl,
and Java for web development [25][26]. Neither study produced a broader model for
comparison. While Jose Lgnacio, Laura Diaz-Casillas, and Carlos A. Lgesias (2008) did
introduce a web framework comparison model, it is limited to non-requirements
comparisons [6]. Therefore, there is currently a need for a comparison model for
choosing the right web framework or CMS.

1.3. Objectives and research questions

In this thesis, I categorize web applications, and generalize common functional
requirements and non-functional requirements accordingly. Based on these efforts, I build
an evaluation model for web frameworks and CMSs. Finally, I analyze well-known web
frameworks and CMS according to the evaluation model. Throughout, I seek to answer
several research questions:
1. How can developers choose which web application framework will fit in a certain
domain?
2. How can they evaluate with reasonable effort which web application framework to choose?

3. Which metrics are applicable to the evaluation?

4. Will the evaluation give meaningful results?

5. How should developers categorize a web application’s domain?

6. Which criteria should be used to compare the frameworks?

1.4. Overall research methodology

Chapters 4 and 5 present case studies for verifying the web application framework evaluation model (WAFEM), applying WAFEM to two popular frameworks: Drupal and Pylon. After evaluating the two frameworks based on the WAFEM, I chose two comparable websites, i.e. two with similar functional requirements and non-functional requirements, one built with Drupal and the other with Pylon. This makes it possible to compare the web application frameworks and CMSs effectively.

This case study is an empirical inquiry that assumes the following:

1. Functional requirements and non-functional requirement are same on both websites.

2. Investigates development costs and development period using Drupal and Ruby on Rails.

3. If two web application frameworks meet all functional and non-functional requirements, the framework with a lower development cost or a shorter development period is the better choice.
Chapter 5 will apply this empirical study to verify that WAFEM can realistically estimate development costs. Estimating these costs ahead of time is necessary to choose the best framework, but we can also reduce development costs by choosing the proper framework. In Chapter 4, we estimate these costs for already finished projects, using the same requirements with Drupal and Pylon. We can verify that this estimate will be valid since we know the actual time and costs of the project. However, Chapter 5 gives a more realistic web framework evaluation because the project is not yet launched.

1.5. Definitions of terms

Since many Internet-related terms are overused and misused, we need to clarify several terms based on academic definitions. First, a framework [3] is a reusable integration of components engineered to facilitate development of a particular type of application. One definition of an application framework is "an integrated collection of object-oriented software components that offers all that's needed for a generic application" [4]. Similarly, a Web Application Framework (WAF) [5] is defined as a set of classes that make up a reusable design for an application or, more commonly, one tier of one application. A Web framework is a collection of software components that support the development and execution of Web-based user interface (UIs): UI that are presented to users at remote locations via their Web browsers [13]. Finally, a Content Management System (CMS) can be defined as a database of information and a way to change and display that information, without spending a lot of time dealing with the technical details of presentation.
We also need to distinguish functional from non-functional requirements. In software engineering, a **functional requirement** defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in **use cases** (WikiPedia).

Conversely, the term **non-functional requirements** [9] has been used to refer to concerns not related to the functionality of the software. “The term “non-functional requirement” is used to delineate requirements focusing on “how well” software does something as opposed to the functional requirements, which focus on “what” the software does.” [11].

**1.6. Assumptions**

With good skilled programmers, every requirement can be met using any of the WAFs: even without access to built-in modules, good programmers can build the required modules given sufficient time. Therefore, every possible WAF can achieve the client’s requirements given unlimited skill and time. For example, say a client wants to build an education application. The client has all the requirements that developers need to fulfill. If the client can choose a PHP developer team, a J2EE developer team, or a Ruby on Rails developer team, which team would the client choose? Since all developer teams meet the requirements by the assumption, the client will choose the team that submits low development costs and a short development period based on the client’s needs. But the
development cost and development period are related, since the development cost will be higher if development period extends. So, to evaluate a WAF it is not sufficient to only investigate users’ requirements.
CHAPTER 2

Conceptual Layouts and Related Research

Since more and more frameworks for distributed applications emerge each year, developers must evaluate the WAFs they are considering, in the context of the project and of the developer or development team. The same developer will not necessarily stick to a framework that he or she knows or has worked with already, because new frameworks may have better benefits than older ones. This suggests several important questions: 1) What is a framework? 2) Do new frameworks keep emerging because the old ones are insufficient? [8] 3) Is there a framework comparison model? 4) If there is such a model, is there a method available to determine which frameworks would be best in the given context? Accordingly, the following sections will review the definition of a framework, the categories of NFRs in application contexts, details about NFRs, and existing work on framework comparison models.

2.1 Conceptual layouts

Web Applications (WAs) are vital company assets, tightly integrated into many business-critical systems: online e-banking, electronic marketplaces, the stock market and e-trading, and electronic access to public administration services are just a few domains that clearly demonstrate WAs’ importance [9]. WAs are mostly built using Web frameworks and Content Management Systems (CMSs). However, many web-developers
misuse or duplicate development terms, which makes understanding web applications even more confusing. People use, for example, web application framework and web framework, or web development platform and web application framework as synonyms. Indeed, there is no clear definition for web frameworks, framework, and CMS. In this section, then, all the terms that this thesis uses are defined.

2.1.1 Web Framework

Fayed defined a framework as a “reusable, ‘semi-complete’ application that can be specialized to produce custom applications,” “in order to reduce the cost and improve the quality of software” [15]. He further divided frameworks into three application frameworks: system infrastructure frameworks, middleware integration frameworks, and enterprise application frameworks. Accordingly, a web application framework is a reusable set of code libraries and tools designed to support the development of web applications. A framework can consist of many parts, including other frameworks and subsystems that are not frameworks [16].

The following table lists some popular web application frameworks, categorized by programming language. Web frameworks are created from the general-purpose language by adding web components.

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Web Application Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP</td>
<td>Zend Framework, CakePhp, symphony, Seagull, etc</td>
</tr>
<tr>
<td>RUBY</td>
<td>Ruby on Rails, Merb, Ramaze, Webby, etc</td>
</tr>
<tr>
<td>Perl</td>
<td>Catalyst, Interchange, Mason, Jifty, etc</td>
</tr>
<tr>
<td>Java</td>
<td>Struts, Hibernate, Java server faces, Spring, etc</td>
</tr>
<tr>
<td>Python</td>
<td>Django, Pylons, Grok, web2py, CherryPy, etc</td>
</tr>
</tbody>
</table>

11
JavaScript – Jquery, Dojo, MooTools, AJS, qooxdoo, etc.

ColdFusion – ColdFusion on Wheels, ColdSpring, onTap, etc

ASP.net—ASP.net MVC framework, DotNetNuke, MonoRail, etc

According to buildwith.com, the PHP language is used most often for web application development (33.45%), followed by ASP.net (24.31%), J2EE (7.79%), and Ajax (7.75%).

ColdFusion includes IDE with CFML, HTML, XML processing, Java, and .NET with ColdFusion Components [Adobe.com]. Perl, which, like PHP, is a general-purpose programming language, is used 1.43% of the time. Based on this chart, not many WAFs are used for web application development.

Some looked at the source code to work out problems, and then used a trial and error approach to get their code running. Others argued that it was too much trouble and they would be better off just writing their code from scratch and not using the framework.

Leesa et al asserted that frameworks consist of three tasks: syntax, semantics, and interfaces [16]. To determine if a framework is suitable for the problem at hand, the software engineer must be aware of the framework’s syntax (which specifies explicitly what the framework does and what functionality it provides) and semantics (which specifies how a software developer actually uses or invokes the framework) [16].

Although implementing anything outside the framework is inherently difficult, web frameworks have many advantages:

1) When using a mature Web framework, a software developer does not need direct contact with the bottom of the API, and needs only write some necessary code. This simplifies the development process, and improves system stability and operational
efficiency.

2) Each mature Web framework has a very professional team to provide full-time work, allowing their respective companies to offer the framework for free to reduce development costs.

3) A well-designed Web framework can provide a distinct system structure, increasing the cohesion of the system. Good structure makes it easier for other people to join the project.

4) An easy-to-use Web framework offers some examples and documentation for users to use optimal practices.

5) The code of mature Web frameworks has been tested often in various application environments, which simplifies the software developers’ code testing process [17].

2.1.2 CMS

In addition to web frameworks, Web Content Management Systems (WCMSs) are significant aspects of e-commerce, organizational, and government portals. WCMSs are considered Content Management System (CMS) software, implemented as a web application, and designed for creating and managing HTML content [26]. CMSs are traditionally used in institutions to allow web content management by people without technical skills, and provide a modern method of web engineering to improve development speed and maintenance effectiveness. The CMS allows individuals to use a template or a set of templates, as well as other tools, to create or modify Web content, e.g. news articles, press releases, papers, photos, graphics, music, video, and presentations [12]. The CMS is considered a static web application showing content on the web.
browser, and can enable users who are not technically proficient to start their own websites.

Using a CMS is a more developer-friendly approach for non-technical web site target users. A good CMS allows a site, or portion of a site, to be integrated and updated without requiring the user to have any special knowledge of HTML or any other programming frameworks. For example, content management systems can be used to update or add product pages in the catalog of an on-line retailer or to add press releases to the site of a growing business. A CMS can be used to allow artists or musicians easily to post new pictures or music on demand, or used by a distributed multi-level sales organization to post the day’s new leads or sales tips to the entire team. Figure 2.1 (taken from builtwith.com) shows CMS distribution for web applications in the top million sites. WordPress, the most used CMS, is an open source blog tool and publishing platform powered by PHP and MySQL. So, this is not a rich client application for business models. By way of comparison, among Enterprise CMSs, Drupal is used 2.27% and Joomla! is used 0.49% of the time.
Even though CMS is still not widely used compared to web frameworks for web application developments, it has many advantages:

1. CMS users can be non-technical people, but can still update and add content to their websites.

2. From the perspective of a technical developer, the developer can give the site owner a full or limited content editing role. Therefore, maintenance costs can be reduced for both the developer and the owner (user).

3. Most CMSs store content in its database with access control mechanisms. So, contents are safer in a CMS because of the built-in security and visibility.

4. Since there are many templates, a developer can concentrate on site content instead of designing software architecture or website features.

5. Once a developer sets up a CMS, the content editor can add or update content from a browser.

Figure 2.1 CMS distribution in Top Million Sites [10].
2.1.3. Functional requirements.

Requirements are a specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system [18]. *Functional requirements* define the internal workings of the software: that is, the calculations, technical details, data manipulation and processing, and other specific functionality that shows how the use cases are to be satisfied. It also contains nonfunctional requirements, which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

More specifically, functional requirements include business requirements, which describe why the organization is undertaking the project [18], as well as user requirements, which describe what the user will be able to do with the product, such as goals or tasks that users must be able to perform. This might include use cases, scenarios, user stories, and event-response tables [19]. Developers might store use cases in a use case document, though some analysts prefer to include their use case descriptions in the software requirements specification (SRS) [18].

A given system could contain only software components, or it could incorporate both software and hardware subsystems. People are a part of a system, too, so certain system functions might be allocated to human beings. Software requirements, then, represent the portion of a system’s functional and nonfunctional requirements that are allocated to software components of the system [18]. Deciding on a web framework
entails finding system requirements that fulfill user requirements. By acknowledging those requirements, one can decide on layers of web frameworks and hardware systems.

2.1.4 Non-functional requirements.

According to Mairiza et al, non-functional requirements can be classified into several categories: Performance, Availability, Usability, Security, Maintainability, Portability, Security, Reliability and so on [20]. Quality attributes are also used to represent NFRs. Mairiza et al emphasized the importance of non-functional requirements, noting that neglecting them “has led to a series of software failures, such as [the] London Ambulance System and New Jersey Department of Motor Vehicles Licensing System” [20]. Generally non-functional requirements are global constraints on a software system. Non-functional requirements are considered hard to model and are stated informally, so it is hard to make them measurable requirements.

There is no standard categorization of non-functional requirements. However, the Microsoft Developers Network (MSDN) suggests a generally accepted categorization for web applications, which I summarize below.

1. Integrity

Integrity defines the consistency and coherence of the overall design. This includes the way that components or modules are designed, as well as factors such as coding style and variable naming.

2. Maintainability
Maintainability is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the functionality, fixing errors, and meeting new business requirements.

3. Reusability
Reusability defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios. Reusability minimizes the duplication of components and also the implementation time.

4. Availability
Availability defines the proportion of time that the system is functional and working. It can be measured as a percentage of the total system downtime over a predefined period. Availability will be affected by system errors, infrastructure problems, malicious attacks, and system load.

5. Interoperability
Interoperability is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems written and run by external parties. An interoperable system makes it easier to exchange and reuse information internally as well as externally.

6. Manageability
Manageability defines how easy it is for system administrators to manage the application, usually through sufficient and useful instrumentation exposed for use in monitoring systems and for debugging and performance tuning.

7. Performance
Performance is an indication of the responsiveness of a system to execute any action within a given time interval. It can be measured in terms of latency or throughput. Latency is the time taken to respond to any event. Throughput is the number of events that take place within a given amount of time.

8. Reliability
Reliability is the ability of a system to remain operational over time. Reliability is measured as the probability that a system will not fail to perform its intended functions over a specified time interval.

9. Scalability
Scalability is the ability of a system to either handle increases in load without impact on the performance of the system, or the ability to be readily enlarged.

10. Security
Security is the capability of a system to prevent malicious or accidental actions outside of the designed usage, and to prevent disclosure or loss of information. A secure system aims to protect assets and prevent unauthorized modification of information.

11. Supportability
Supportability is the ability of the system to provide information helpful for identifying and resolving issues when it fails to work correctly.

12. Testability
Testability is a measure of how easy it is to create test criteria for the system and its components, and to execute these tests in order to determine if the criteria are met. Good
testability makes it more likely that faults in a system can be isolated in a timely and effective manner.

13. Usability

Usability defines how well the application meets the requirements of the user and consumer by being intuitive, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience.

Design qualities represent the design of the software application, including considerations of architecture, modularity, packaging, extensibility, and so on. Run-time qualities explore all of the system aspects that are directly involved with system dynamics such as performance, scalability, transaction integrity, security, and fault tolerance [24]. System qualities provide helpful information and how well isolate the software from tests. User qualities explain user interface design with the consumer in mind, so that the interface is intuitive to use. Figure 2.2 illustrates these qualities.
2.1.5 Requirements according to their application domains.

Each business type requires different use cases with different applications. Each use case’s functional requirements are different for each use, though there are some common requirements (such as login access for web applications, database interface for any applications, and so on). Therefore, each software domain has different software
functional requirements. Here, we are specifically looking at web software applications; each web application also has its own domain, such as e-commerce web applications, search engine web applications, school register web applications, web game applications, education application, e-mail applications, video conference applications, etc. Given that new internet applications are emerging continuously, this list of domains will no doubt continue to expand.

Jeff Offutt mentioned in his paper “Web Applications Quality Attributes” that security, reliability, availability, and scalability are important in web based applications, as well as functionality [25]. Along with performance and maintainability, these are also the most common non-functional requirements for web applications. Based on these elements, I want to identify web applications along with business scales and business types, and prioritize each quality attribute according to its business types and scales. For example, Table 2.1 shows that each of the categorized web sites has different non-functional requirements, estimates the priority of those non-functional requirements, and shows total.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Financial Web Applications (Bank, Stock)</th>
<th>Search Engines (Google, Yahoo)</th>
<th>E-commerce (apple.com)</th>
<th>Enterprise portal (osu.edu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Availability</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Usability</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Security</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Maintainability</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Reliability</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Scalability</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>27</td>
<td>27</td>
<td>26</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 2.1 Application domain and NFRs [7]
In conclusion, since each web application has different functional and non-functional requirements, the Web architecture designer has to consider each framework’s and CMS’s non-functional requirements. Choosing frameworks or CMSs is one of the major decisions a web developer makes, and will influence a lot of subsequent development choices.

2.2 Previous framework comparison models.

Many informal comparisons of web development platforms or frameworks exist. Most of them are based on the authors’ subjective opinions, without having tried out other web frameworks. Those models do not give any help for choosing a web application in other domains. As noted earlier, some studies have made direct comparisons of frameworks in a certain domain. Laakso and Niemi, for instance, compared Java-based web frameworks fitted for their project. But all the previous work in this area is limited to comparing frameworks for certain specific projects, such as Gerdessen’s assessment for Everest’s application [8]. Gerdessen compared Sun Blueprint and Spring Framework, both Java frameworks, in the domain of customizability and modifiability. Peng Wang [14] compared four Java frameworks in the technical domains of navigation rules, validation, internationalization, type conversion, IoC support, and post and redirect. Jose Ignacio, Laura Diaz-Casillas, and Carlos A. Iglesias compared Ruby on Rails, Grails, Trails, and Roma in the areas of presentation, security, usability, testing, service orientation, component orientation, and adoption in the domains of data migration, ORM (object-relational mapping), validation, and transaction [5].
Laakso and Niemi did show some systematical framework analysis, using the GQM (Goal, Question, Metric) model [24]. GQM analyzed goals derived from questions about the application and found answers to the questions (the metric). For example, if one web application is set, developers can ask what the web application is for. Those are mainly functional and non-functional requirements. If one realizes functional and non-functional requirements, decision criteria can be evaluated in the AJAX-enabled Java-Based Web Application Frameworks Laakso and Niemi analyzed. Casagni and Lyell exhibited similar evaluation structures. Their comparison structure consists of (1) Design Property Features, (2) Framework Impact Features, and (3) Quality Attributes. They compared J2EE and MAS (a FIPA-compliant multi-agent system) in terms of three sets. The first set of the elements is composed of language structure such as language size, inheritance, encapsulation, and coupling. These evaluation structures are well known as programming language comparison methods. The second set of the elements comprises packing, deployments, component access, and design pattern usage. The third set of the elements contains quality attributes comparisons. Casagni and Lyell compared the two frameworks with aspects of similarity and difference in all three sets of elements.

Unfortunately, none of these studies can answer a common question among developers: “What framework do I have to use for the best results, given the web application my company wants to build?” This is because all the comparison models compared functional and non-functional requirements of the web frameworks. Since each project has its own requirements, we cannot say one web framework is better than another for any given project. Each paper showed only limited comparisons, such as two
quality attribute-comparisons between two web frameworks, even though development teams face many more choices.
CHAPTER 3
Comparison Model for Web Application Frameworks

3.1. Introduction

When a web development team develops a project, there are two important factors: how skillful the development personnel are, and what kind of tools they use. These two factors determine the development period. For this thesis, how well a project manager leads the project team and what kind of software development process is practiced are both considered invariable. This is because management skill and software development processes are not comparable factors to compare web frameworks. Because of a new framework, a development team can complete a project faster with their existing management and development process. One might think that developing a project using a new web framework might take more time than using an already known framework. But if the new framework is easy to use, the development team can learn it and still finish the project faster. Therefore, we need to compare learning cost and leverage of using the web framework. For example, a new framework may require more learning time for a given team, but if the new framework covers more functional and non-functional requirements, it could actually reduce development costs.

Application tools are considered for both web frameworks and CMSs. Assuming the same level of developer skill, good web frameworks shorten the development period,
in part by providing components from the web framework, help centers, and mature discussion groups. Additionally, compatibility with other tools and adoptability to other languages can shorten development time. In other words, both human skills and web frameworks decide the development time. Since longer development time increases development cost, and since lower cost determines the better choice given the same output, skilled developers and the proper web framework can thus reduce project costs as well.

Figure 3.1 Development process
Figure 3.1, shown above, shows a typical software development life cycle. At the time the development team starts to make a web framework decision, the team knows most of the functional requirements and some of the non-functional requirements, by requirement analysis, including business analysis. After these analyses, the analyst chooses the web framework for each layer [17]. Picking web frameworks can thus affect the costs of designing overall software architecture, and implementation of the non-functional and functional requirements. Before implementation, of course, developers need to know the web frameworks. If the developer does not know the frameworks’ base languages and their usage, he or she must learn them, increasing learning costs before the team can implement functional and non-functional requirements according to the signed software architecture. So, at the point in time of deciding on a framework, lead developers clearly need a web framework decision model to help them make a better choice.

**System Costs**

1. Installation Cost
2. Costs of Tools

**Design Costs**

**Learning Costs**

**Implementation Costs**

1. Functional requirement Costs
2. Non-functional requirement Costs

**Total cost = System Costs + Design Costs + Learning Costs + Implementation Costs**
Evaluating web application frameworks is not about achieving users’ requirements, but rather evaluating how fast a project can be built with low cost. If there are several candidate web frameworks, we can estimate the relative development costs according to the candidate frameworks.

3.2. Problem statement

There are too many web frameworks, platforms, and languages available in the marketplace, and it is extremely difficult for companies or their development teams to make cost-effective, timely, and objective evaluations of these technologies. Moreover, there are no methods available to determine which frameworks would be best in a given context; there are no real methodologies for framework comparison at all [8]. To go over all the web frameworks is too broad, while comparing only two frameworks does not give much help to the company that wants to decide which web framework—and accordingly, which application software architecture—is the best fit for a specific context. All web frameworks have their own strengths and weaknesses. For example, WordPress is known as an open source blog software package that works exceptionally well as a content management system. It makes it easy for anyone to setup and manage a website, without having any advanced technical skills, and would be the best tool for a simple blogging website. Then, developers or users could customize the site using CSS, HTML, and JavaScript. If the purpose of the blog is to present one’s thoughts or artworks, WordPress would be the most effective web framework, compared to a customized blog webpage with its own URL.
In this example, based on the publisher’s purposes, WordPress would be the best framework. But for more complex situations, we need to trace how to compare web frameworks, and what the measures (standards) are for such comparisons. In previous research, some quality attributes are compared in certain contexts. However, applying this methodology to web frameworks raises a problem. For example, one candidate web framework might take 10 seconds to perform, while the other takes 30 seconds. The first one could be the better web framework if we compare only performance. But if the latter one takes 20 days to deploy while the first one takes 2 months to deploy, we cannot automatically decide that the first one is better. Therefore, the previous comparison models do not give any specific comparison for a certain project because they all compare only the quality attributes of the web frameworks.

3.3. Assumptions

My assumption is that all the project requirements can be met by all efforts within a certain period of time no matter what framework is chosen or CMS, drawing on human experience and development tools. First, even if developers are not experienced in a certain language, if they have a longer period of development time, they will complete the requirements. Similarly, skilled developers will meet the functional and non-functional requirements with any kind of web framework or CMS. In other words, under this assumption anyone with any web application development tool can meet the requirements. For instance, a novice programmer may have to learn the Java language before starting the project, which would of course cause higher development costs due to the longer time, but even in that scenario the developer could implement the project from
scratch. The problem, then, is not completing the project but completing it with the lowest costs.

3.4. Analysis of Web Framework

3.4.1 System Costs

This system costs are fundamental for installing the environmental settings before development. Once we chose a web framework, we need to install and configure the various frameworks. For example, Drupal WCMS needs MySQL, PHP, and Apache Server to be installed. And each installation needs proper configuration: the configuration of MySQL is provided in a file named my.cnf, PHP in php.ini, and Apache in httpd.conf. Likewise, under J2EE with Struts 2, we need to configure web.xml, and struts.xml for the Eclipse IDE.

Further, not all frameworks are freeware. If a developer chose to do .NET development along with ColdFusion, the company would have to pay for both software suites. So, the cost includes the price of software, installation, and configuration costs. For projects integrating several web frameworks with one another, configuration can be complicated, e.g. the attribute names of tables in the database needs to be aligned with the names of the class objects in the application code.

3.4.2 Design Costs

After deciding on the web framework, we need to design the architecture of the software architecture according to the requirements of a project. This software architecture also has to be aligned with the architecture of the chosen web frameworks. Design costs include the designing of the interfaces, classes, and design patterns used in
the application. If the project requires a database—most projects do—we have to consider the costs of designing the database such as tables and schemas.

The software architecture determines how the functional requirements as well as the quality attributes of the projects, such as compatibility, extensibility, modularity, reusability, robustness, and usability, are met. If more non-functional requirements are to be considered, we need more time to design to fulfill the non-functional requirements. However, if the chosen web framework already has built-in capabilities that the project needs with respect to functional requirements, we do not need to design those functionalities. For example, Python with Pylon has a built-in pop-up login function (Figure 3.2), like those found on many websites. So, if one of the project requirements happens to be a pop-up login, one would not need to design the pop-up login because the web framework already offers it.

![Figure 3.2 Pop-up login Functionality in Python with Pylon.](image)

If all functional requirements are built into the web frameworks, system costs are the only costs for the project since the frameworks pay design costs and implementation costs for the functional requirements. However, the non-functional requirements of web
frameworks are not necessarily directly related to the non-functional requirements of the project. For example, if reusability is configured in a project, reusability of the web framework affects the reusability of the project. In the worst case, the web framework may have to be restructured or redesigned for better modularity, reusability, extensibility, and maintainability of the project. The following lists represent samples of the non-functional requirements of web frameworks which relate to the design cost of the project. The design qualities shown in Fig 2.2 are related to a given project’s design costs.

**Design Qualities of a web framework**

- **Integrity**
- **Reusability**
- **Extensibility**
- **Maintainability**
- **Modularity**

Good design quality for non-functional requirements of web frameworks will save time to enable the possibility of using functionalities in a module without code refactoring [5]. Therefore, well-designed non-functional requirements are related to reduced or eliminated design costs on a project. In the case of poor design qualities, a software architect may need to refactor the architecture of the web framework so that developers can use the sub-functionalities in the web framework. In that case, good design quality can contribute to good integrity, maintainability, and reusability in the project. However, even with the best design qualities, software design using the web
frameworks may be poor: the overall design quality depends on the design qualities of the project, not of the web frameworks.

**Run-time quality of a web framework**

- Availability
- Interoperability
- Manageability
- Performance
- Reliability
- Scalability
- Security

Even if the chosen web framework has good performance and if performance is one of the non-functional requirements, the performance of the project again depends on the design performance of the project. Run-time quality is in this way the same as design quality. If the performance of web framework is not satisfactory, we need to refactor the structure of the web framework’s performance.

**User qualities**

- Usability (Learning Costs)
- Understandability (Learning Costs)

**System Qualities**

- Supportability (Learning costs)
- Testability (Testing costs)
User qualities and system qualities are not design costs but learning costs. These two non-functional requirements are explained in further detail in section 3.4.3.

In summary, design costs are composed of the amount of classes, interfaces, and design patterns applied to functional and non-functional requirements that are not provided by web frameworks. This is because if the functional requirements are not provided by web frameworks, application developers have to design the functional architecture of the requirements. Design qualities and run-time qualities are non-functional requirements that affect design costs. If the non-functional requirements in design qualities and run-time qualities are not provided by web frameworks, application developers have to design the requirements’ architecture for the non-functional requirements. Similarly, if non-functional requirements in design qualities and run-time qualities are provided by web frameworks but the quality is not satisfactory given the requirements, developers have to restructure the architecture of the non-functional requirements.

3.4.3 Learning Costs

Learning cost is the time taken to understand the appropriate language or tool. Human factors affect application development in important ways. For this study, I assume fixed software development methodologies, such as waterfall or agile. With this fixed development model and fixed number of development team members, the variables are the development teams’ skill levels in the framework and the base language. With skilled expertise, any web application can be made. With non-skilled expertise, any web application can be made too, but with a different development period. Any development
team can likewise meet its functional requirements, but their project completion times vary. As new developers develop a project, they can meet all the requirements, but the project period will be prolonged. The below graph shows that as developers in the development team have more skill in a particular language, the development period is shortened. So the previous assumption is that any developer can meet the all the requirements at the cost of project duration. If the main language is easy to use, the developer’s learning curve will be high.

A web application project has its own functional and non-functional requirements, as does a candidate web framework. Ignacio et al evaluate Ruby on Rails, Grails, Trails, Roma according to eight aspects that four web frameworks provide to developers [5]. The user of the web framework is a web application developer, and the user of the web application is an end user. Quality attributes of the web frameworks affect the learning costs. The following schema shows which quality attributes of a web framework affect learning costs in a project overall.

**User Qualities**

**Usability of a web framework (learning costs)**

If usability of a web framework is good, application developers will get used to the web framework sooner and will find it easier to adopt and use the web framework for the project. If a web framework has good usability, developers learn how to use it with low cost, compared to a web framework that is difficult to use. Good usability can make for a stiff learning curve.

**System Qualities**
Supportability of a web framework (learning costs)

If the web framework has a mature help group or discussion group, the developer will pass some obstacles in no time. Therefore, learning costs to overcome technical difficulties is low if the web framework has strong supportability, and development time will accordingly be shortened.

3.4.4 Implementation Costs

The project requirements can be met using any web framework, assuming a fixed level of developer skill. But the proper web framework can reduce the development period using the same skill level. The period to fulfill the project requirements is thus only dependent on its web frameworks.

Benefit of functional requirements = built-in components of functional requirements

Benefit of non-functional requirements = built-in components of non-functional requirements

Costs of functional requirements = number of functional requirements – benefit of built-in components of functional requirements in a project.

Costs of non-functional requirements = number of non-functional requirements - benefit of built-in non-functional requirements.

For example, if the web application requires login functions, using a web application framework with login components will save development time compared to frameworks
without them. Based on previous domain analysis, the decision maker needs to analyze what functionality the web application needs (functional requirements), as well as its non-functional requirements, and check how many built-in components match with requirements. If more matching components exist, the development period will be shorter.

3.4.5. Testing Costs

**System Qualities**

**Testability of a web framework (testing costs)**

After one phase of a web application development is complete, the software development life cycle shows that the last phase is testing. If developers can test the web frameworks using testing tools, the developer will reduce the cost of testing the web application because the web framework is part of the web application. The testing aspect of a web framework is therefore related to cost of a web application, and good testability can produce low development costs.

3.5. Methodologies

<table>
<thead>
<tr>
<th>Parameter</th>
<th>System Cost</th>
<th>Design cost</th>
<th>Learning costs</th>
<th>Implementation Costs</th>
<th>Testing Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FR costs</td>
<td>FR costs</td>
</tr>
<tr>
<td>Web Frameworks</td>
<td>Installation costs</td>
<td>Cost of classes</td>
<td>Usability Supportability Manageability Understandability Adoption</td>
<td>FR costs</td>
<td>FR costs</td>
</tr>
<tr>
<td>Costs of tools</td>
<td></td>
<td>Cost of Interfaces</td>
<td>Cost of Database</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

38
The following shows a sample questionnaire to survey the cost of a project using a certain web framework. According to the development process, we already know the project requirements.

**System costs:**

How difficult is the setting the development environments?

How many software programs do you need to install for the web framework?

How much does it cost to buy the web frameworks/WCMS?

**Design costs:**

Do you need to design classes?

Do you need to design interfaces?

Do you need to apply a design pattern?

Do you need to design a database?

**Learning costs:**

Usability: how easy is it to use the framework/WCMS?

Adoption: how popular is the framework/CMS?

Supportability: how well organized is the user group or help group?

Manageability: how easily can the content editor add the contents (with any involvement of technical developers)?

**Implementation costs:**

**Non-functional requirements**

How many non-functional requirements does the web framework/WCMS not cover?

Security: how mature is the web framework/WCMS?
Extensibility: can this exchange and communicate with external systems written and run by external parties?

Performance: how responsive is this web framework/CMS?

**Functional requirements**

How many functional requirements does the web framework/WCMS not cover?

These may include login, roles separation, theme templates, scheduling module, e-commerce, etc, depending on the target application.

**Testing costs:**

Testability: Does the web framework support a testing framework?

Testability: Does this web framework/WCMS have debugging tools?

For each question, the framework can be evaluated from 0 to 5. If project requirements are different with same web framework, then design costs and implementation costs change. If the web framework is changed with the same project, system costs, testing costs, design costs, and implementation costs change.
CHAPTER 4
Case Study: P4 Application

4.1. Introduction

In this chapter, we will estimate the cost of a web application using WCMS and web frameworks. The P4 project was finished with Python with Pylon, as shown in Figure 4.1, and later the decision was made to use Drupal CMS. So this project was making a new Drupal P4 application to meet the same project requirements. Figure 4.2 shows that new P4 application using Drupal CMS. The first developer used Pylon because the backend application used Python, and because Pylon was the perfect web framework integrated with Python.

Figure 4.1 P4 web application using Pylon
4.2. Requirements Analysis

This web application was created to deliver the Calico application running in Linux. A user inputs a MOL file and downloads an output PDF file. Inside the web application, the P4 executes the Calico Python application, analyzes the input data, and show the result of the input file with graphical output. A user has to login to use the analysis application through an LDAP server, as shown in Figure 4.3.

![Figure 4.2 P4 Use case](image-url)
Figure 4.3 Functional requirement for uploading MOL input file.

Figure 4.4 Functional requirement for pasting sample mol data.
Figure 4.5 Previous file storage system shows an input file and an output file.

Functional requirements:

The usecase shown above explains the functional requirements. Additionally, the P4 system has to keep track of old input file and output files, so that users can check and download previous inputs and output files.

Non-functional requirements:
We do not need to consider scalability because a few tens of students (at most) are expected to use the P4 web application. Maintainability is also not necessary because the system does not add any more content, e.g. no daily updates. Performance does not need to be considered because most of the calculation requires the Calico application running in Linux. The configured non-functional requirements are security, testability, and extensibility.

4.3 Cost analysis with Drupal CMS

SYSTEM COSTS: 2/10

How difficult is setting up the development environments? (0-5) 0
Installing PHP, SQL, Apache server separately (0-5) 2
How much does it cost to buy the web framework/WCMS? (0-5) 0

DESIGN COSTS: 2/20

How many classes do you need? (0-5) 2
How many interfaces do you need? (0-5) 0
How many different frameworks have to be integrated? (0-5) 0
Do you need to design a database? (0-5) 0

LEARNING COSTS: 7/20

Usability: how easy is it to use/learn the framework/WCMS? (0-5) 1
Supportability: how well organized is the user group or help group? (0-5; 0=the best help group, 5=no help group) 2
Manageability: how easily can the content editor add content (with any involvement of technical developers)? (0-5; 0=easiest) 1
Adoption: How popular is the framework/CMS? (0-5; 0=most popular, 5=not popular) 2

**IMPLEMENTATION COSTS: 19/35**

**Non-functional requirements: 7/10**

Security: how mature is the web framework/WCMS? (0-5; 0=most mature) 2.

Extensibility: can this framework exchange and communicate with external systems written and run by external parties? (0-5; 5=no extensibility) 0.

**Functional requirement 12/25**

Does Drupal support the module for running a Python executable file? (0-5) 5

Does Drupal support the login module? (0-5) 0

Does Drupal support the LDAP Access module? (0-5) 2

Does Drupal support the folder access module? (0-5) 5

Does Drupal support the theme for presentation? (0-5) 0

**Test Costs: 5/5**

Testability: Does this web framework/WCMS have debugging tools? (0-5) 5

<table>
<thead>
<tr>
<th></th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Costs</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Benefits</td>
<td>8</td>
<td>18</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1 Cost Analysis developing P4 web application using Drupal
Table 4.1 shows that the total cost for building the P4 web application is 90. Drupal can save a cost of 61. If Drupal is chosen, 29 tasks can be done for the developer, meaning more than half of the developments are done if we choose Drupal CMS.

4.4 Cost analysis with Pylon.

Every requirement is same as in 4.2, except P4 Pylon uses a local database for login instead of using the LDAP server like Drupal. Therefore, LDAP access functional requirements are removed from this application.

**SYSTEM COSTS: 1/15**

How difficult is setting up the development environments? (0-5) 1

How much does it cost to buy the web framework/WCMS? (0-5) 0

**DESIGN COSTS: 7/20**

How many classes do you need? (0-5) 4

How many interfaces do you need? (0-5) 0

How many different frameworks have to be integrated? (0-5) 2

Do you need to design a database? (0-5) 1

**LEARNING COSTS: 14/20**

Usability: how easy is it to use/learn the framework/WCMS? (0-5) 3

Supportability: how well organized is the user group or help group? (0-5; 0=the best help group, 5=no help group) 3

Manageability: how easily can the content editor add content (with any involvement of technical developers)? (0-5; 0=easiest) 5

Adoption: How popular is the framework/CMS? (0-5; 0=most popular, 5=not popular) 3
IMPLEMENTATION COSTS: 16/35

Non-Functional requirements: 5/10

Security: how mature is the web framework/WCMS? (0-5; 0=most mature, 5=least mature) 2.

Extensibility: can this framework exchange and communicate with external systems written and run by external parties? (0-5) 3

Functional requirement 10/20

Does Pylon support the module for running a Python executable file? (0-5) 3

Does Pylon support the login module? (0-5) 1

Does Pylon support the folder access module? (0-5) 3

Does Pylon support the theme for presentation? (0-5) 3

Test Costs: 1/5

Testability: Does this web framework/WCMS have debugging tools? (0-5) 1

<table>
<thead>
<tr>
<th></th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Costs</td>
<td>1</td>
<td>7</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Benefits</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>10</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.2 Cost analysis: developing P4 web application using Pylon
Table 4.2 shows that the total cost for building the P4 web application is 85. Because the P4 application using Pylon has no requirement to access an LDAP server, the total costs of functional requirements are reduced to 20 from 25. Pylon can save a cost of 47. If Pylon is chosen, 38 tasks can be done for the developer, reducing the needed effort by more than half.

4.5. Assessments and evaluations

Tables 4.3 and 4.4 show summaries for development costs when using Drupal and Pylon. Here, the Cost Reduction Ratio (CRR) = Benefits of using a web framework / Total Costs. In this analysis, 1 is full cost and 0 is no cost for developments. The CRR for Drupal is 61/90 = 0.67. 67% of development efforts and costs can be saved if we use Drupal, compared to Pylon’s CRR of 47/85=0.55 (55% savings). Therefore, Drupal can save more effort and cost compared to Pylon for the P4 project.
<table>
<thead>
<tr>
<th>Costs</th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Costs</td>
<td>Installation costs 2/5</td>
<td>Cost of designing classes 2/5</td>
<td>Cost of designing interfaces 0/5</td>
<td>Cost of designing database 0/5</td>
<td>Cost of integrating frameworks 0/5</td>
<td>Usability 1/5</td>
</tr>
<tr>
<td>Benefits</td>
<td>8</td>
<td>18</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.3 Summary for P4 Application Development Costs using Drupal
<table>
<thead>
<tr>
<th></th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functionality</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Non-functionality</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Installation costs</td>
<td>1/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs of tools</td>
<td>0/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs of class</td>
<td>4/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs of interfaces</td>
<td>0/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs of design database</td>
<td>1/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs of integrating</td>
<td>2/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>frameworks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supportibility</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manageability</td>
<td>5/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adoption</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>1/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storing input files</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and output files</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Presentation</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>2/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extensibility</td>
<td>3/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testability</td>
<td>1/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Costs</td>
<td>1</td>
<td>7</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Benefits</td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>10</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.4 Summary of P4 Application Development Costs using Pylon

The Drupal CMS has a lower development cost compared to Pylon because Drupal installations include a fully functional web application. However, Pylon gives structure to classes and interfaces, and details the implementation that needs to be created according to the project. In many cases, if Drupal gives the functionalities for the web project, the class and interface are not necessarily created. Instead, each functional module can be installed and used in the web application. Therefore more design cost can be saved with Drupal compared to a web framework. In the implementation cost, we
again saved more cost using Drupal because Drupal offered many functional modules relevant to our project. In the case of LDAP access, Drupal provided an LDAP module, but we had to modify the code because the module does not provide member attributes login. It could check the CN (Common Name) only. We had to implement the file system to show and download previous inputs and outputs manually, in both Drupal and Pylon. Testability costs for Drupal are 5 out of 5 because Drupal does not have a testing module, and it is very difficult to test the contents because its contents are stored in the database.
CHAPTER 5

Case Study: PFast Application

5.1. Introduction

This web application was proposed to deliver the PFast desktop application, using a .NET to web service. PFast was running in Windows, and uses Excel files for both input and output, calculating and analyzing the input data and showing the results with graphical output. Users have to login via an LDAP server to use the application. As with the P4 application, here there is no need to consider scalability because only a few tens of students would use the application. Maintainability is also unnecessary because a customer adds inputs from the PFast site and gets the downloadable result from the same site.

Figure 5.1 P5 web application using Drupal
5.2. Requirements Analysis

As described in 5.1, this web application delivers the PFast Windows application. The PFast web application has same requirements as P4, except its deploying web is in Windows instead of Linux.

![Figure 5.2 PFast Usecase](image)

**Figure 5.2 PFast Usecase**

![Figure 5.3 Functional requirements for uploading an Excel input file.](image)

**Figure 5.3 Functional requirements for uploading an Excel input file.**
Figure 5.4 Previous file storage system shows an input file and an output file.

Functional requirements:

The usecase shown above explains the functional requirements. Additionally, PFast has to keep track of old input file and output files, so that the user can check and download previous inputs and output files.

Non-functional requirements:

In addition to the requirements mentioned in 5.1, maintainability is not necessary because system does not add any more content. Performance also does not need to be considered because most of the calculation takes Calico Application running in Linux. The configured non-functional requirements are security, testability, and extensibility.
5.3 Cost analysis with Drupal CMS

**SYSTEM COSTS: 2/10**

How difficult is setting up the development environments? (0-5) 0
Installing PHP, SQL, and Apache server separately (2)
How much does it cost to buy the web framework/WCMS? (0-5) 0

**DESIGN COSTS: 2/20**

How many classes do you need? (0-5) 2
How many interfaces do you need? (0-5) 0
How many different frameworks have to be integrated? (0-5) 0
Do you need to design a database? (0-5) 0

**LEARNING COSTS: 7/20**

Usability: how easy is it to use/learn the framework/WCMS? (0-5) 1
Supportability: how well organized is the user group or help group? (0-5; 0=the best help group, 5=no help group) 2
Manageability: how easily can the content editor add content (with any involvement of technical developers)? (0-5; 0=most easily) 1
Adoption: How popular is the framework/CMS? (0-5; 0=most popular, 5=not popular) 2

**IMPLEMENTATION COSTS: 19/35**

*Non-Functional requirements: 7/10*

Security: how mature is the web frameworks/WCMS? (0-5; 0=most mature) 2
Extensibility: can this framework exchange and communicate with external systems written and run by external parties? (0-5; 0=a lot, 5=no extensibility) 0
**Functional requirements: 12/25**

Does Drupal support the module for running a Windows executable file? (0-5) 5

Does Drupal support the login module? (0-5) 0

Does Drupal support the LDAP access module? (0-5) 2

Does Drupal support the folder access module? (0-5) 5

Does Drupal support the theme for presentation? (0-5) 0

**Test Costs: 5/5**

Testability: Does this web framework/WCMS have debugging tools? (0-5) 5

<table>
<thead>
<tr>
<th></th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Costs</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Benefits</td>
<td>8</td>
<td>18</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.1 Cost Analysis developing PFast web application using Drupal

Table 5.1 shows that the total cost for building the PFast web application is 90. Drupal can save a cost of 61 if the developer. If Drupal is chosen, 29 tasks can be done for the developer, nearly half. Here, the total and development costs are identical to the P4 web application using Drupal, because PFast’s requirements are the same.

**5.4 Cost analysis with Groovy on Gails**

Every requirement is same as in 5.2, so the total costs are the same: 90. In the case of P4, Drupal and Pylon have different total development costs for LDAP access, as
described in 4.3. But for PFast, Drupal and Groovy on Grails are analyzed using the same project requirements.

**SYSTEM COSTS: 3 out of 15**

How difficult is setting up the development environments? (0-5) 3

How much does it cost to buy the web framework/WCMS? (0-5) 0

**DESIGN COSTS: 4 out of 20**

How many classes do you need? (0-5) 3

How many interfaces do you need? (0-5) 0

How many different frameworks have to be integrated? (0-5) 1

Do you need to design a database? (0-5) 0

**LEARNING COSTS: 11 out of 20**

Usability: how easy is it to use/learn the framework/WCMS? (0-5) 2

Supportability: how well organized is the user group or help group? (0-5; 0=the best help group, 5=no help group) 3

Manageability: how easily can the content editor add content (with any involvement of technical developers)? (0-5; 0=most easily) 4

Adoption: How popular is the framework/CMS? (0-5; 0=most popular, 5=not popular) 2

**IMPLEMENTATION COSTS: 16 out of 35**

**NON-Functional requirement 4 out of 10**

Security: how mature is the web frameworks/WCMS? (0-5; 0=most mature) 3

Extensibility: can this framework exchange and communicate with external systems written and run by external parties? (0-5; 0=a lot, 5=no extensibility) 1
**Functional requirement 12 out of 25**

Does Groovy on Grails support the module for running a Python executable file? (0-5) 4

Does Groovy on Grails support the login module? (0-5) 0

Does Groovy on Grails support the folder access module? (0-5) 5

Does Groovy on Grails support the theme for presentation? (0-5) 2

Does Groovy on Grails support the LDAP Access module? (0-5) 1

**Test Costs: 2 out of 5**

Testability: Does this web framework/WCMS have debugging tools? (0-5) 2

<table>
<thead>
<tr>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>Costs</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Benefits</td>
<td>7</td>
<td>16</td>
<td>9</td>
<td>13</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5.2 Cost Analysis developing PFast web application using Groovy on Grails

Table 5.2 shows that total cost for building the PFast web application is 90. Groovy on Grails can save a cost of 54. If Groovy on Grails is chosen, 36 tasks can be done for the developer. Groovy on Grails can reduce development efforts by more than half.
5.5. Assessments and evaluations

Tables 5.3 and 5.4 show summaries for development costs, using Drupal and Groovy on Grails. As above, the Cost Reduction Ratio (CRR) = Benefits using a web framework / Total Costs. In this analysis, 1 is full cost and 0 is no cost for developments. The CRR for Drupal is \( \frac{61}{90} = 0.67 \): 67% of developments efforts and costs can be saved if we use Drupal. The CRR for Groovy on Grails is \( \frac{54}{90}=0.6 \), a 60% savings. Therefore, if we compare PFast developments using Drupal and Groovy on Grails, Drupal can save more effort and cost for the PFast project.

<table>
<thead>
<tr>
<th></th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Functional requirements</td>
<td>Non-functional requirements</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

|                         | Installation costs 2/5 | Costs of tools 0/5 | Cost of designing classes 2/5 | Cost of designing interfaces 0/5 | Costs of designing database 0/5 | Cost of integrating frameworks 0/5 | Usability 1/5 | Supportability 2/5 | Manageability 1/5 | Adoption 2/5 | Login 0/5 | LDAP access 2/5 | Storing input files and output files 5/5 | Information Presentation 0/5 | Running a Windows executable file 5/5 | Security 2/5 | Extensibility 0/5 | Testability 5/5 |
| Costs                   | 2            | 2           | 6             | 12                  | 2         | 5           | 29          |
| Benefits                | 8            | 18          | 14            | 13                  | 8         | 0           | 61          |

Table 5.3 Summary for PFast Application Development Costs using Drupal
In this case study, we implemented PFast with Drupal. Now we want to use the comparison model to compare the development costs to Groovy on Grails. The costs of Drupal are experienced values but the costs in Groovy on Grails are surveyed costs. But the survey was given exclusively to an expert on Groovy on Grails. Groovy on Grails is a web framework that gives frames for fast developments, providing many built-in components. If we design PFast using Groovy on Grails, we would have to design classes for the project in order to use the functional built-in components. Since these projects

Table 5.4 Summary for PFast Application Development Costs using Groovy on Grails

<table>
<thead>
<tr>
<th>Costs</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>36</td>
<td>54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Costs</th>
<th>System cost</th>
<th>Design cost</th>
<th>Learning cost</th>
<th>Implementation Cost</th>
<th>Test Cost</th>
<th>Total Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Costs</th>
<th>Installation costs 3/5</th>
<th>Costs of class 3/5</th>
<th>Costs of interface 0/5</th>
<th>Costs of design database 0/5</th>
<th>Cost of integrating frameworks 1/5</th>
<th>Usability 2/5</th>
<th>Supportability 3/5</th>
<th>Manageability 4/5</th>
<th>Adoption 2/5</th>
<th>Login 0/5</th>
<th>Ldap access 1/5</th>
<th>Storing input files and output files 5/5</th>
<th>Information Presentation 2/5</th>
<th>Running a Windows executable file 4/5</th>
<th>Security 3/5</th>
<th>Extensibility 1/5</th>
<th>Testability 2/5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Costs</td>
<td>37</td>
<td>4</td>
<td>11</td>
<td>12</td>
<td>4</td>
<td>2</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
have few requirements, designing interfaces is not necessary. So, designing classes and implementing classes are the heaviest costs for Groovy on Grails, compared to Drupal.
CHAPTER 6

Conclusion

In the previous sections, we have proposed a web framework comparison model. It is based on the definition of common web application development processes. Parameters were grouped into a set of aspects that summarized the general features and issues found in the web application development process. The method followed in these case studies was relatively informal, with ordinal ratings being collected for my projects, P4 and PFast, and from one Groovy on Grails developer. More systematic data collection techniques could be adopted, to estimate the benefits and the costs more precisely. However, several variables are difficult to quantify (e.g. supportability, usability).

The model has been applied to three web frameworks: Drupal WCMS, Pylon, and Groovy on Grails. These frameworks are popular because they ease the complexity and relieve development efforts. However, to take full advantage of a web framework’s benefits, necessary studies must be done to find out the optimal web framework to apply to one’s project. The main purpose of this thesis is to help technical managers to find the best-suited web frameworks through this comparison model. In order to calculate the cost estimation, several steps are taken:

System Costs

1. Installation cost
2. Costs of Tools

**Design Costs**

**Learning Costs**

**Implementation Costs**

1. Functional requirement Costs
2. Non-functional requirement Costs

**Total cost = System Costs + Design Costs + Learning Costs + Implementation Costs**

Similarly, the overall benefits of adapting a particular framework can be quantified:

System benefits = total system costs - system costs using the chosen web framework
Design benefits = total design costs – design costs using the chosen web framework
Learning benefits = total learning costs – learning costs using the chosen web framework
Implementation benefits = total implementation costs – implantation using the chosen web framework

**Total benefits = System benefits + Design benefits + Learning benefits + Implementation benefits**

**Cost Reduction Ratio (CRR) = Benefits using a web framework/ Total Costs.**

I applied this cost evaluation model to the P4 and PFast projects, which have similar requirements, using Drupal CMS, Pylon, and Groovy on Grails. As shown above, overall Drupal can maximize the benefits for both projects. Specifically, Drupal can save learning costs, design costs, and implementation costs because Drupal can offer ease of
use and many useful modules for a user. But web frameworks such as Pylon and Groovy on Grails offer ease of development and many components for developing. The difference is that web frameworks have to be developed to meet requirements with the provided components. Drupal, though, can use the module without implantation of any classes and interfaces.

However, if the requirements are not provided by Drupal modules, the learning costs, design costs, and implementation costs can easily become maximized. This is because development costs can be overloaded through following Drupal developmental rules. If requirements are needed to customize Drupal modules, implantation costs can be higher because developers have to change the implementation of the module. In the case of the P4 and PFast projects, we needed to change the modules.

Future work in this area could go in several directions:

- The cost estimation needs to be applied to a larger survey population.
- The cost estimation model needs to be applied to more complicated projects, i.e., those with more customized functional and non-functional requirements.
- More empirical research on survey questionnaires related to system costs, design costs, learning costs, implementation costs, and testing costs is required.
REFERENCES


2. Joseph Amrith Raj, “Middleware 101”

3. Eric gamma, Richard Helm, Ralph Johnson, and John Vlissides, “Design Patterns Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995


