

EE productivity report 2011 [Tur10], these following seven most widely-used frameworks are selected for further research and analysis:

- Spring MVC (SpringSource.org)[WB05]
- Java Server Faces (J2EE and JSR Standard)[Man05]
- Wicket (Apache Software Foundation)[DH09]
- Seam (JBoss)[All08]
- Struts2 (Apache Software Foundation)[BDS08]
- Tapestry (Apache Software Foundation)[Shi04]
- Stripes (Stripes)
- The existing system, pure JSP and Servlet

2.3. General Web Framework Criteria and Frameworks Analysis

Even though, web frameworks give you many benefits, but there are several criteria that should not be neglected. These are the example of those criteria:

2.3.1. Testability

Testability is the degree to which an artifact or module support testing in a given test context. Testability cannot be measured directly. In this research paper, the testability of the framework measured roughly by analyzed the architecture and test supported tools provide by the frameworks community. For instance, MVC architecture or other frameworks, which supports clearly separation of each layer, or some frameworks might provide powerful testing API. These two examples provide improvement of testability for the framework.

2.3.2. Learning curve

The lower the learning curve is, the better quality the framework is. A good web framework must be not just powerful, productive, flexible, but also need to be easy to understand by the developer, who is new to the framework.

2.3.3. Community and support

The supports from the framework's community is also important. The community such as forum, web board, mailing list are good source to find the answer, when the question about framework is raised. In a vibrant community, there is higher chance for an inexperienced developer to get help and guidance from the users, who experienced using the framework.

The word, support, also includes framework development team and resources (books, web-based tutorial) . A good framework should have an active development team. It is normal to have small bugs in the framework, but the framework development team should get rid of those bugs as soon as reported. Books are the best resource for fully understanding of the mechanism and architecture of the framework. Also, the online resources for the framework should be search engine friendly.

2.3.4. Tools and IDE

With supported tools and IDE, the system development using the framework can be a lot more easier. IDE supports can be a famous Java IDE, Eclipse, with specific framework plug-in. The example of what are these plug-ins capable of is auto-completion, drag and drop user interface components, code generation, XML configuration file editor, and test case generation. Another example of supported tool is Maven Archetype. Maven can established the project structure and handles all dependency of the modules.

2.3.5. Architecture and patterns reflected on the framework

The idea of this criteria is to analyzed the core structure of the framework. Nevertheless, some architecture and pattern might clearly improve the existing system, or even cure the problem, considered as a plus. For example, Model-View-Controller architecture (MVC) provides clear separation of concern, which leads to ease of understanding, and testability. On the other hand, any frameworks, which use the Command pattern on the presentation layer will caused the same problem as the proposed architecture.

2.3.6. Configuration method and complexity

Almost every web framework needs configuration, which effect directly to the complexity and learning curve of the framework. Lower configuration is easier to manage and understand, which means the framework might loses the flexibility because, the framework handles most of the process, and the developer cannot overwrite those processes in the simple way.

2.3.7. Amount of artifacts produced by the framework and the degree of complexity

This determines the amount of artifacts produced by the framework under the same circumstance. The lower amount reflects less complexity of the overall system, but lower amount of artifacts usually come with the higher degree of complexity. XML configuration file is one of a good example. Frameworks based on XML configuration file is flexible, scalable, and produced less artifacts, but in exchange, higher degree of complexity, harder to maintain and debug replaced.

The following diagram shows the comparison result of the frameworks selected from the 1st framework selection, in the context of some part of the criteria explained in previous section [Figure 2.6]. The full version of the comparison table is in the appendix of this research paper.

	Configuration Complexity	Learning Curve	Testability	Community & Support	Tools & IDE
Spring MVC	-	-	+	+	-
JSF	+	0	0	+	+
Wicket	+	+	+	+	0
Seam	0	-	+	+	+
Struts2	0	0	0	0	+
Tapestry	+	-	-	-	0
Stripes	+	+	+	-	-
Existing System	+	+	-	+	-

Figure 2.7.: General Web Framework Criteria Comparison

Symbols	Description	Score
+	The framework shares the identical characteristic to the good quality web framework in context of the criteria.	2
0	The framework reflects some characteristic of the good quality web framework in context of the criteria. Not completely identical, but not too bad.	1
-	The framework has no trace of good quality web framework characteristic and does not give any supports related to the criteria.	0

Figure 2.8.: Conversion table for [Figure 2.6]

After research through each framework in context of the general web framework criteria, the result measured by convert symbols into score [Figure 2.7], indicates that Wicket, Java Server Faces, Seam, and Struts2 have most identical

characteristic to good quality web framework [Figure 2.8].

The result from full version of the comparison table indicates that these, seven frameworks and the existing system can be put into three different categories based on the performance of those frameworks in context of focusing criteria: good, medium, and low quality frameworks. Wicket, JSF, Seam, Stripes, and Struts2 are the first category, good quality frameworks. All of them share these following similarity:

- All of them have clear separation between presentation and business logic. Three of them, which is Struts2, Stripes, and JSF are MVC architecture frameworks.
- Most of them provides method to handles the code redundancy of the system. Two of them are component-based framework [PRV05], which provides high reusability for user interface components, while Struts2's core concept, *Interceptor*, provides high reusability of the common code in the system.
- Most of them are flexible enough to supports the integration with other frameworks or technology. Seam, itself, is a full-stack framework, which provides multiple alternatives frameworks or technology to fit in each stack. Struts2 supports integration with many view options such as JSP/JSTL, Tiles, Velocity, FreeMarker, Excel, XSL, PDF, while JSF does not provides any integration support, but there are many 3rd party user interface component libraries, which is ready to use and powerful.
- Most of them have low learning curve and configuration complexity. JSF, Stripes and Wicket have low learning curve. JSF requires very simple configuration xml file and annotation, Stripes need only few lines of annotations and all Action Beans are discovered by the framework automatically, while Wicket requires no configuration at all (zero configuration). Struts2 and Seam requires a bit of an effort than Wicket and JSF, but still, less than other frameworks
- Most of them have good community and supports. Wicket and Seam have active community, while JSF and Struts2 are widely used. Stripes has a serious problem about the community and supports. Stripes is not actively developed, small community, poor documentation, and no books.
- Most of them have good IDE and tools supported. Seam, JSF, and Struts2 are widely used and there are many tools and IDE support, while Wicket and Stripes does not need any specific tools and IDE.
- JSF has a serious issue about high memory consumption.

The existing system is a medium quality example. First, the existing system is not a framework. It is pure JSP and Servlets implement Command pattern.

	Total Score	Comments
<i>Wicket</i>	9	
<i>JSF</i>	8	<i>Performance Issue</i>
<i>Seam</i>	7	
<i>Struts2</i>	6	
<i>Stripes</i>	6	<i>Small community, not actively developed</i>
<i>Existing System</i>	6	<i>Already implemented</i>
<i>Spring MVC</i>	4	
<i>Tapestry</i>	3	<i>No backward compatibility</i>

Figure 2.9.: Comparison result summary

The architecture is very simple and required only basic knowledge of web development to understand. Only web.xml need to be taken care of. No specific IDE and tools support needed.

Spring MVC and Tapestry have poor result and can be categorized into low quality frameworks. Spring MVC is one of the most well-known web framework in its high flexibility, high capability of integration to other technology and frameworks, and ease of testing (which is originally, the purpose of Spring MVC). However, Spring MVC also known-well as one of the most complex framework. With very steep learning curve and pure XML configuration method, the flexibility of Spring MVC causes extremely high complexity single XML configuration file with no tools and IDE support. Tapestry has the lowest score and crucial drawbacks, which is no backward compatibility. Until the latest release (Tapestry 5), the framework developer team still have not promised anything for the next version.

2.4. Requirement Gathering

The requirements from the users of the MeDIC and XAM system, who get benefits from this technical report is needed in order to select the suitable frameworks. The requirements meeting was held in a meeting room of the building where all the attendances are working at January 2011, 28th. There were 5 attendances. All of the attendances are developers working in the MeDIC or the XAM project.