

The frameworks with IDE and tools supported for testing such as test case generation, or provides powerful testing API are advantageous.

2.5.5. 5th priority (5)

- Model-based generation support

Since there is a project currently running about generation of web-based prototypes for business applications, If the structure of the prototype after integrates to framework has static pattern and possible to generate using code generator theory, would be advantageous.

2.6. Summary

In previous sections of this technical report, seven famous frameworks are introduced. The figure below [Figure 2.10] shows the comparison of each framework in context of the requirements.

Frameworks/ Requirements	1	2	3	4
<i>Spring MVC</i>	✗	✗	✗	✓
<i>JSF</i>	✓	✓	✓	✗
<i>Wicket</i>	✓	✓	✗	✓
<i>Seam</i>	✓	✓	✓	✓
<i>Struts2</i>	✓	✗	✓	✓
<i>Tapestry</i>	✓	✓	✓	✗
<i>Stripes</i>	✓	✗	✗	✓
<i>JSP/Servlet</i>	✓	✗	✗	✓

Figure 2.11.: Frameworks comparison against requirements

For the common pitfalls (1st priority requirement), almost of the frameworks passed, except Spring MVC and Stripes. Spring MVC has serious problem with very high configuration complexity (pure XML, configuration over convention), and Stripes has a small community and not actively developed.

For the component-based, inheritance structure architecture, and the request delegation supported (2nd requirement), all of the frameworks have Portlet API, so the Inheritance supported structure is possible. However, four frameworks have component-based architecture, which are JSF, Seam, Wicket, and

2. Background

Priority	Requirements
1	Common Pitfalls : <ul style="list-style-type: none">• Avoid complex and hard to understand configuration files.• Many small files are more preferable than less amount of large files.• AJAX is optional but the system should be able to work without it.• The framework's community should be active and the project must be currently running.
2	<ul style="list-style-type: none">• Component-based, inheritance supported architecture.• Delegation and navigation support.
3	<ul style="list-style-type: none">• Good IDE support.
4	<ul style="list-style-type: none">• Easy to test.

Figure 2.12.: Requirements Summary

Tapestry. Also, delegation and navigation supported can be achieved by Business Process Management framework integration such as jBPM. However, only Spring MVC and Seam supports perfect integration with jBPM, and only JSF and Struts2 have navigation rule. The navigation rule controls the flow of page navigation. Seam might be the most direct solution of this requirement since, Seam provides both navigation rule and jBPM to control page flow and business process flow.

Spring MVC and Stripes have serious problem with poor IDE support (3rd priority requirement). Even Wicket and Stripes has no official IDE support, but since Wicket composed of HTML templates and pure Java, and Stripes composed of JSP and pure Java, any Java IDEs and HTML editor provides full support. Other frameworks passed this requirement especially, JSF and Seam.

After requirements analysis, three frameworks were chosen to implement the prototypes. Those three frameworks are Wicket, JSF, and Struts2.

Originally, Seam were chosen instead of JSF because, it fulfils all requirements. Seam is a full-stack framework which integrates several frameworks together. For presentation layer, Seam provides variety choices including JSF and Wicket. There are several advantages of Seam using JSF over stand alone JSF such as:

- JSF and EJB naturally, not aware of each other at the framework level because, these 2 frameworks are design as separate frameworks with their own architecture and does not intentionally, develop to integrate together. Seam make these 2 frameworks realize each other by provide a consistent, annotation-based approach. EJB business components in Seam will be used directly to back JSF web forms or handle events without Manage Bean.
- Seam fixes the most crucial weakness on memory consumption of JSF. Using Seam, JSF memory consumption is optimized.

- Seam provides seamless integration with many open-source frameworks such as jBPM, Drools, JBoss Portal, Maven.
- Seam provides better AJAX support with two open source JSF-based AJAX solutions: ICEfaces and AJAX4JSF. Seam even provides a special layer called JavaScript remoting layer where AJAX clients can call the server-side components directly without any additional layer between.
- JSF follows J2EE standard, but not follows Java EE 5.0 simplify programming model. Seam provides full-stack architecture and follow Java EE 5.0 simplify programming model.
- Seam reduces Boiler-plate code and code redundancy by replacing Manage Beans with transactional components directly, simplified navigation rules, and configuration by exception.
- Seam provides unique technique, "Dependency Bijection" (Dependency Injection and Dependency Outjection). When view method invoked, Seam checks for the annotation for injection and inject the objects first and as soon as the method finish the execution, Seam will check for the annotation for Outjection and expose the objects.
- Greatly improve ORM performance and support.
- More web friendly such as support web transaction, support multiple browser windows/tabs, support back button, fix JSF URL problem, and etc.
- Enhance security such as authentication, fine-grain access control, rule-based instance-level access control. Also include security provides by Drools.
- Easier to test with Seam Testing framework. Easy test outside the container.
- One of the most powerful Seam tool is Seam Project Generator (Seam Gen). Seam gen serves two main functions: setup Seam-based project and required environments, create an application prototype by reverse engineering a database schema and all of the necessary artifacts to create, retrieve, update, delete data in the database (CRUD application generation). Seam gen provides quick start for the framework starter, solves complexity of environment setup, and reduces time consuming in CRUD applications generation.

Due to conflict integration between Seam environment and existing environment which will be explained in detail later in Chapter 3, the prototype implementation using Seam was failed. Therefore, JSF was selected for prototype implementation instead of Seam.

2. Background

Wicket were chosen because, it fulfills all requirements except IDE/tools support and some outstanding properties related to the requirements. For example:

- Wicket has very simple component-based architecture and complete separation of presentation layer and business logic layer which makes Wicket project easy to test. Wicket Tester is also a very powerful testing library. (4th priority requirements)
- There are no configuration file even annotation which means there is no configuration complexity at all. Wicket's community is well-known as one of the most active community (1st priority requirement)

Even Wicket doesn't have any official tools supported, but the architecture of Wicket project composed of POJOs and HTML templates so normal IDEs, which supports Java and HTML would be acceptable (3rd priority requirement). For example, Dreamweaver, Eclipse, Netbeans.

Other frameworks are not qualified. Spring MVC fails most of the requirements especially, almost first 3 priority requirements. Tapestry was precluded because of serious problem about no backward compatibility supported for old versions. Stripes fails 1st and 3rd priority requirements because, the project are not actively developed and no official IDE supported. Since all other frameworks are precluded, Struts2 is the best choice amongst the frameworks left.