Enterprise Software Platforms Report

Project: Enterprise Software Platforms Report

Project team: BBB-BC

Team: Joeri Berman (2834499)

Luc Urlings (3071081)

Stijn Baltessen (3064565)

Wouter Vandenboorn (3099571)

Floris Feddema (3188256)

Davy de Haas (3099695)

Opdrachtgever: Jeffrey Cornelissen

Version: 0.2

Version Date: 17 februari 2020

Status: Concept

# Table of Contents

[**Table of Contents**](#_f16f2vftvt4q) **2**

[**Introduction**](#_cnjl504ela4d) **3**

[**1. Selection Criteria**](#_blpod8882pkd) **4**

[**2. Select Choices**](#_xabhd451nswz) **5**

[Framework](#_xzidf2y1p5cr) 5

[Spring (Java)](#_eiu20lf6jmk7) 5

[Struts (Java)](#_g82ozfsnzgn7) 5

[.NET Framework (C#)](#_3kqjdcjxsyoz) 5

[.NET Core (C#)](#_6qmjf4w29duz) 5

[**3. Evaluation Matrix**](#_btuhjez1s4od) **6**

[**4. Advice**](#_wv7mfssbvmo1) **7**

# 

# Introduction

In this report, we will explore multiple enterprise software platforms to find the best suitable platform possible for the case study. We are going to think what characteristics of a platform are important in an enterprise context and what is important to the organisation and its end customers.

We will make advice for the case study about what we think is the most context-fitting enterprise software platform, including supporting frameworks.

# 1. Selection Criteria

In this chapter, the selection criteria will be discussed.

1. Costs of the product: This is one of the most important factors. The most expensive parts are probably the developers for specific languages and frameworks.
2. License constraints: Some languages, frameworks or developing tools require a license to be used in enterprise applications
3. Maturity of the product: The stability and amount of features of a framework or language.
4. Reliability: Well tested and bugfree.
5. Active community: The size and the amount of activity of the framework or language.
6. Documentation: the elaboration of the documentation.
7. Learning curve: The difficulty of the language or framework.
8. Previous experiences: The previous experiences of the developers that will be working on the project.
9. Scalability: if it is possible to easily scale up the product
10. Load support (users, requests, data, volume): if it can be used by a high amount of users or not without becoming unusable
11. Security: The security vulnerabilities in the framework or language.
12. Exception handling: The exception handling of the framework or language
13. Life expectancy: Will the language be supported or will it be deprecated in the foreseeable future.

# 2. Select Choices

## Framework

### Spring (Java)

Spring is an application framework and inversion of control container for the Java platform. By using certain extensions, it is possible to use the Java EE (Enterprise Edition) platform. It can be used by various scenarios, including a large bank that needs to rewrite the whole codebase.

Testing is made easy in Spring due to the Spring Dependency Injections. This helps developers to insert test data by, for example, mocking.

Spring has a very flexible configuration. Developers can choose either XML or Java-based annotations for configurations, thus making the jobs of developers a lot simpler.

The AOP module makes it possible to have different compilation units or a separate classloader per developer.

### Struts (Java)

Struts is an open-source framework for Java EE interactive web applications. It uses the Java Servlet API to encourage developers to adopt a model-view-controller architecture (MVC).

Struts works efficiently with SOAP, AJAX and REST API’s and it uses simple POJO actions.

A lot of integration possibilities with other frameworks via plugins and is well tested and stable.

### .NET Framework (C#)

.NET Framework is a software framework developed by Microsoft and runs primarily on Microsoft Windows. It is used to build Windows desktop apps and large-scale enterprise applications. It has third-party libraries, NuGet packages and technologies that are not yet available in .NET Core, but it is less high performance and will be deprecated in the near future.

### .NET Core (C#)

.NET Core is a cross-platform and open-source framework developed by Microsoft, which makes it possible to develop applications on any platform. It is often used for cloud applications or refactoring large enterprise applications into microservices. Core is highly scalable, has a high performance and can easily be containerized.

# 3. Evaluation Matrix

Below you find the chosen products and its scores to the defined selection criteria.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria:** | **Weight:** | Spring | Struts | .NET Framework | .NET Core |
| **1** | 7 | 1.0 | 0.7 | 1.0 | 0.8 |
| **2** | 5 | 0.6 | 0.6 | 1.0 | 1.0 |
| **3** | 4 | 0.8 | 0.6 | 1.0 | 1.0 |
| **4** | 7 | 0.9 | 0.7 | 0.8 | 1.0 |
| **5** | 5 | 0.9 | 0.7 | 1.0 | 0.9 |
| **6** | 5 | 0.9 | 0.6 | 1.0 | 0.8 |
| **7** | 3 | 0.6 | 0.6 | 0.6 | 0.6 |
| **8** | 2 | 1.0 | 0.0 | 0.8 | 0.8 |
| **9** | 7 | 1.0 | 0.8 | 0.8 | 1.0 |
| **10** | 5 | 0.7 | 0.6 | 0.8 | 1.0 |
| **11** | 10 | 1.0 | 1.0 | 1.0 | 1.0 |
| **12** | 4 | 1.0 | 1.0 | 1.0 | 1.0 |
| **13** | 5 | 1.0 | 0.8 | 0.3 | 1.0 |
| Total |  | 61.8 | 50.1 | 60.1 | 64.5 |

1. Costs of the product
2. License constraints
3. Maturity of the product
4. Reliability
5. Active community
6. Documentation
7. Learning curve
8. Previous experiences
9. Scalability
10. Load support (users, requests, data, volume)
11. Security
12. Exception handling
13. Life expectancy

# 4. Advice

According to the matrix, the .NET Core framework is the best one to use within an enterprise platform.