FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Comparison and Evaluation of JavaScript Preprocessing Languages

Masterstudium:
Software Engineering & Internet Computing

Manuel Mertl

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übsetzer
Betreuer: Ao. Univ. Prof. Dipl.-Ing. Dr. Franz Puntigam

Problem Statement

A lot of different JavaScript preprocessing languages can be found on the Web nowadays.

Therefore, choosing the appropriate language for a certain problem can be a very difficult decision.

If the choice is not made with little forethought, it might lead to possibly serious problems later in the projects.

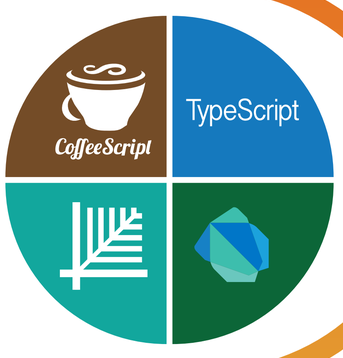
Many of these problems could be avoided, if developers had a way of systematically evaluating alternatives.

Research Question

Which specific JavaScript preprocessing language

CoffeeScript, TypeScript, LiveScript, Dart

is best suited for a new web application project, depending on some of the projects properties and its environment?



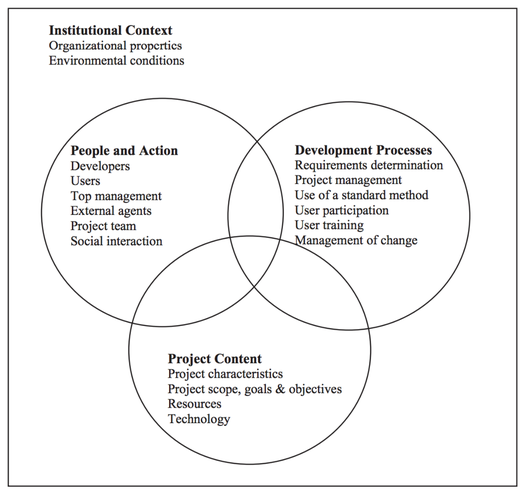
Methodology

1 Determining Important Project Properties

We only concentrate on the Project Content and People & Action area, becase we do assume that the domains of Development Process and Institutional Context have no direct influence on the choice of the preprocessor. They are the same no matter which language we choose.

Project Content
- Project Characteristics
- Resources
- Technology

People & Action
- Developers
- Project Team



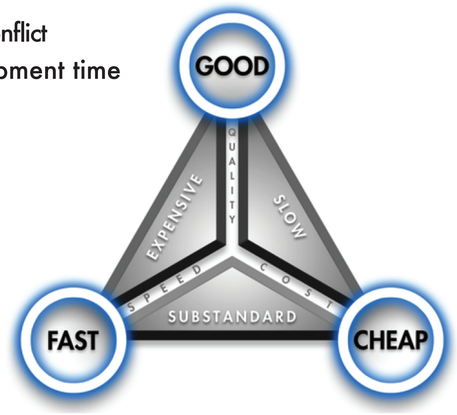
2 Finding the Related Pre-Processing Language Features

Here we find the pre-processing language features that are related to the important project properties given in the previous step.

Goals of the Management
- minimize the financial resources
- decrease the development time

Goals of the Development Team
- increase the code quality
- increase the given development time

Shared Conflict
- development time



3 Measuring this Features

After evaluating and comparing each feautre accross our four investigated pre-processing languages, we get the following cost-benefit analysis template:

Group of interest: Management

	CoffeeScript	TypeScript	LiveScript	Dart	Weighting
Financial Resources					
Developers					
Costs of employed developers	3	4	1	2	[X]
Costs of freelance developers	2	4	3	1	[X]
Availability of developers	4	2	1	3	[X]
Risks					
Popularity	4	2	1	3	[X]
Trends	2	4	1	3	[X]

Group of interest: Development Team

	CoffeeScript	TypeScript	LiveScript	Dart	Weighting
Code Quality					
Language Features					
Support for static typing	0	2	2	1	[X]
Classes	1	2	1	2	[X]
Interfaces	0	1	0	1	[X]
Inheritance and mixins	1	2	2	2	[X]
Generics	0	1	0	1	[X]
Modularization	1	1	2	2	[X]
Operator overloading	0	0	0	1	[X]

Group of interest: Management and Development Team

	CoffeeScript	TypeScript	LiveScript	Dart	Weighting
Development Time					
Language Features					
Code Reuse	1	3	2	4	[X]
Tool Support					
IDE Support	2	3	1	3	[X]

Conclusion & Future Work

Though the answer to our research question depends very much on the used weighting factors, the result of our two example analyses show that Dart and TypeScript might be the best option to choose, followed by CoffeeScript and LiveScript.

Since the cost-benefit analysis template can be seen as the main contribution of this thesis, it would be kind of interesting to expand this template in some future work in several ways. E.g.:

- expand number of features
- divide given features into subfeatures
- add new preprocessing languages

This given suggestions would enhance the cost-benefit analysis template and help providing better results.

References

L. Prechelt.
An empirical comparison of seven programming languages.
Computer, 33(10):23–29, Oct 2000.

Laurie McLeod and Stephen G. MacDonell.
Factors that affect software systems development project outcomes: A survey of research.
ACM Comput. Surv., 43(4):24:1–24:56, October 2011.

Benjamin C. Pierce.
Types and Programming Languages.
The MIT Press, 1st edition, 2002.

Layard Richard and Stephen Glaister.
Cost-benefit analysis.
Cambridge University Press, 1994.

Kontakt: manuel.mertl@gmail.com