

SOFTWARE CHALLENGE

Wettkampfsystem - Gegenwart und Zukunft

4. April 2014

Sven Koschnicke



GLIEDERUNG

1. Aktueller Stand

2. Probleme

3. Vorgehen

4. Erhaltung

5. Aufwand

AKTUELLER STAND

AKTUELLER STAND

- Sprache: Ruby 1.8
- Framework: Ruby on Rails 2.3

AKTUELLER STAND

- Sprache: Ruby 1.8
- Framework: Ruby on Rails 2.3
- ca. 12.000 Zeilen Code

AKTUELLER STAND

- Sprache: Ruby 1.8
- Framework: Ruby on Rails 2.3
- ca. 12.000 Zeilen Code
- Test coverage: 0%
- Dokumentation: Wenig
- über mehrere Jahre von wechselnden Entwicklern erstellt

PROBLEME

RUBY 1.8

RUBY 1.8

- end of life Juli 2013

RUBY 1.8

- end of life Juli 2013
- keine security fixes mehr

RUBY 1.8

- end of life Juli 2013
- keine security fixes mehr
- aktuelle Version ist 2.1

RAILS 2.3

RAILS 2.3

- end of life Juni 2013

RAILS 2.3

- end of life Juni 2013
- keine security fixes mehr

RAILS 2.3

- end of life Juni 2013
- keine security fixes mehr
- aktuelle Version ist 4.1

ZUSÄTZLICH

Beheben von Fehlern und Veränderungen schwierig.

VORGEHEN

1. Umstellung auf Ruby 2.1 & Rails 4.0



1. Umstellung auf Ruby 2.1 & Rails 4.0
2. Anwendung wieder lauffähig bekommen



1. Umstellung auf Ruby 2.1 & Rails 4.0
2. Anwendung wieder lauffähig bekommen
3. Automatische Tests ermöglichen



1. Umstellung auf Ruby 2.1 & Rails 4.0
2. Anwendung wieder lauffähig bekommen
3. Automatische Tests ermöglichen
4. minimale Dokumentation erstellen



NÖTIGE VERBESSERUNGEN

Verbesserungen, die Arbeit bei der Migration sparen:

1. Übersetzungen vereinheitlichen



NÖTIGE VERBESSERUNGEN

Verbesserungen, die Arbeit bei der Migration sparen:

1. Übersetzungen vereinheitlichen
2. Darstellung vereinheitlichen



NÖTIGE VERBESSERUNGEN

Verbesserungen, die Arbeit bei der Migration sparen:

1. Übersetzungen vereinheitlichen
2. Darstellung vereinheitlichen
3. Kontext-Problematik lösen

ERHALTUNG

Vorgehen für zukünftige Arbeiten definieren:

1. Ticket in Issuetracker
2. Tests schreiben
3. Änderung implementieren
4. Dokumentation
5. Code-Review mit Mitarbeiter
6. Deploy auf Staging
7. Manueller Test
8. Ticket schließen

AUFWAND

AufgabeWochen

Ruby 2.1 & Rails 4.0

✓

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	½

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	$\frac{1}{2}$
Darstellung vereinheitlichen	4

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	$\frac{1}{2}$
Darstellung vereinheitlichen	4
Kontext-Problematik lösen	4

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	$\frac{1}{2}$
Darstellung vereinheitlichen	4
Kontext-Problematik lösen	4
Anwendung wieder lauffähig bekommen	8

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	½
Darstellung vereinheitlichen	4
Kontext-Problematik lösen	4
Anwendung wieder lauffähig bekommen	8
minimale Dokumentation erstellen	2

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	½
Darstellung vereinheitlichen	4
Kontext-Problematik lösen	4
Anwendung wieder lauffähig bekommen	8
minimale Dokumentation erstellen	2
Arbeitsabläufe für die Zukunft definieren	1

Aufgabe	Wochen
Ruby 2.1 & Rails 4.0	✓
Automatische Tests ermöglichen	✓
Übersetzungen vereinheitlichen	½
Darstellung vereinheitlichen	4
Kontext-Problematik lösen	4
Anwendung wieder lauffähig bekommen	8
minimale Dokumentation erstellen	2
Arbeitsabläufe für die Zukunft definieren	1
ca. 5 Monate	