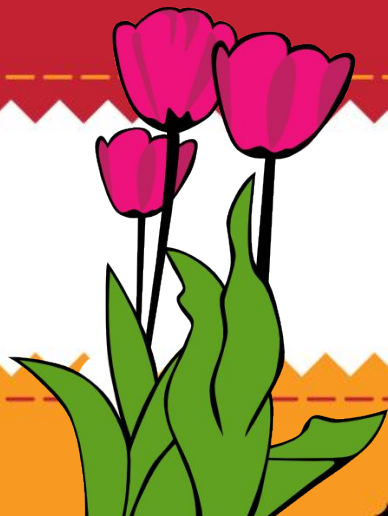
Политика невмешательства

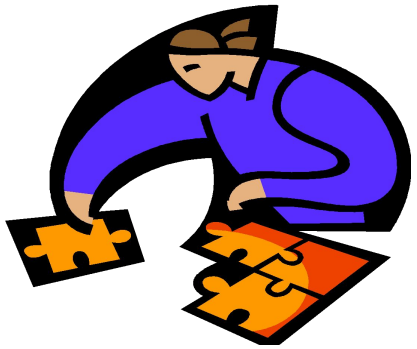
Как в tulp.ru решена задача сохранения изменений состояния системы

DevConf



http://www.devconf.ru

Постановка задачи



- Состояние системы
  - Модель A
  - + Модель B
  - + Модель C

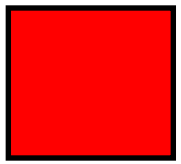
- Требуется просматривать
  - состояние "до" изменения
  - состояние "после" изменения
  - служебную информацию (кто, когда, ...)

Варианты решения

- Inline code
- Callback
  - before_save, after_save
- Observer
  - paper_trail, act_as_versioned
- Decorator
  - method_decorators
- Aspect
  - aspectr, aquarium

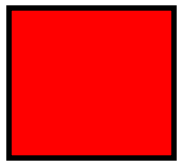
Inline code

```
save_before( a, b, c );
change( a, b, c );
save_after( a, b, c );
```

- Загрязняется основная логика
- Необходимо поддерживать изменения
-  (не катит)

Callback

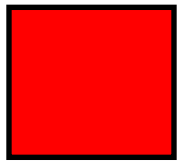
```
before_save   :save_state_before
after_save    :save_state_after
```

- Сохранение состояния 1 модели (не группы)
- Дополнительный код в моделях
- 

AR::Observer:

обычная реализация

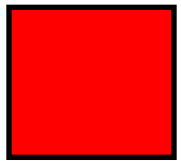
```
class UserObserver < ActiveRecord::Observer
  def after_save(user)
…
  end
end
```

- Сохранение состояния 1 модели (не группы)
- 

AR::Observer:

реализация через gem

```
class User < ActiveRecord::Base
  # команда подключения функционала гема
  has_paper_trail
  # acts_as_versioned
end
```

- Сохранение состояния 1 модели (не группы)
- ~~Дополнительный код в моделях~~
- 

Decorator

```
class UserService < MethodDecorators
```

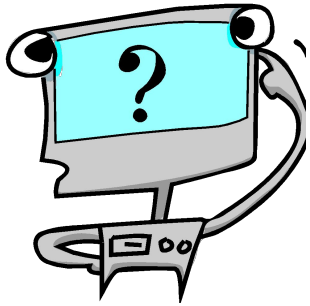
gem install method_decorator

+Log_user_state

```
  def approve(user_id)
    User.find(user_id).approve()
    Billing.add_user(user_id)
    Complaint.scope_user(user_id).each(&:close)
    save!
  end
end
```

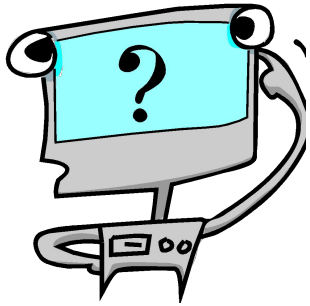
- Нужно много различных декораторов
- [redacted]де есть следы (хоть и приемлемо мало)
-

Aspect



Кто знает,
что такое АОП?

Aspect



- AOP
- Aspect
- Advice
- Join Point

Программа

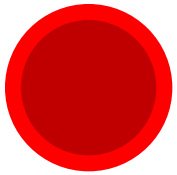
```
…
…
user.set_role("admin")
…
…
```

Aspect

```
class UserAspect < AOP
  advice User,  :set_role, options
end
```

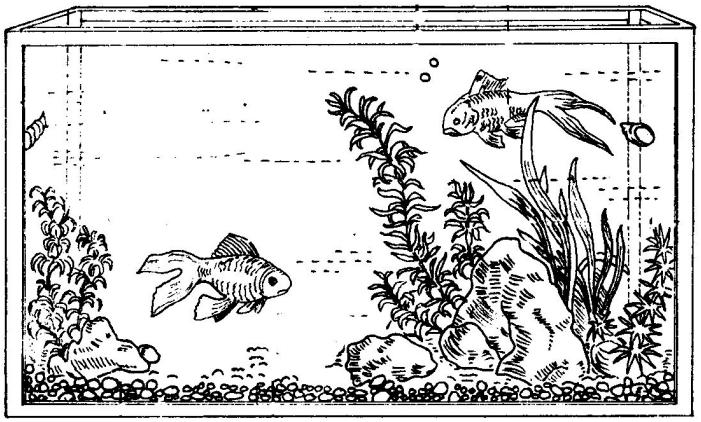
Aspect

```
class Changelog
  watch UserService, [:approve, :disapprove], {
    state_before: lambda{ |user_id| … },
    state_after:  lambda{ … },
    state_error:  lambda{ … }
  }
end
```

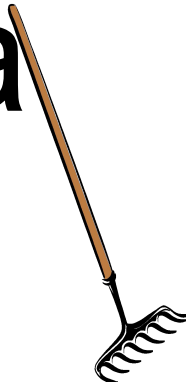
- Нет ни единой строчки в моделях/сервисах/etc
- Все настройки – в одном месте
- 

Выбор

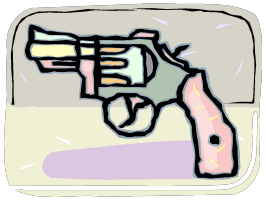
- AspectR (2002)
- Aquarium - `gem install aquarium`



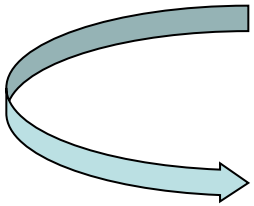
Практика



- Aquarium
  - мощный
  - слишком



- Упрощение интерфейса
  - более жёсткие требования
  - обработка ошибок
  - спокойный сон после release



Откат

в жизни модератора

- Детальная трассировка изменений
- У меня все ходы записаны (с) 12ст
- Легко откатить изменения
- Долой коррупцию, распил и откаты!

Ссылки

http://bit.ly/cR400e - AOP @ wikipedia

http://bit.ly/15q8bT - Decorator @ wikipedia

http://bit.ly/It8d7l - method_decorators

http://bit.ly/KXu9ZA - aquarium @ github

http://bit.ly/rUyqf - aquarium @ rubyforge

ВСЕМ

Спасибо за внимание