Samvera - Keeping Up To Date

Insert Self Deprecation Slide Here

Who is this clown?

Rob Kaufman
@orangewolf
rob@notch8.com
https://www.notch8.com

Founder of Notch8 - An App Development Consultancy since 2007
This Deck
http://bit.ly/n8sc2018-2

When Should We Upgrade

How Often Considerations

Estimating Upgrades

Ruby Version

Rails Version

Gemfile Updates

Test Versions

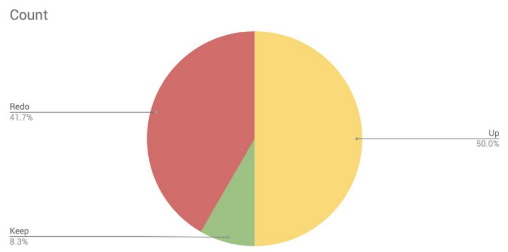
Test Coverage

Brakeman Results

App Size Comparison

Develop a Plan

We'll need to update or replace each gem used. Fortunately it is a small list, so that will help. Several gems didn't make it into the modern day and we'll need to replace those. Colors below denote High, Medium or Low risk.



Package	Locked	Requirement	Latest	Recommend
haml		3.1.18	5.0.3	Update, Syntax Changes
gnuplot		2.6.2	2.6.2	Keep
giraffesoft-resource_controller	0.61	0.61		Replace
cancan	1.1.1	1.1.1	cancancan 2.0.0	Update

Exercise - Tools

Brakeman

Bundler Audit

RetireJS

rails stats

Brakeman

```
gem install brakeman
brakeman -o brakeman.html -f html
```

Bundler- Audit

In Gemfile copy the following line:

```
gem 'bundler-audit', git:
'https://github.com/notch8/bundler-audit ',
groups: [:development, :test]
```

On command line

```
git checkout -b maintenance
bundle
bundle-audit check --update
-o=tmp/audit.html
```

RetireJS

```
npm install retire
retire -j app -n node_modules --outputformat
text --outputpath results.txt
```

Rails Stats

```
rails stats
```

Server Side - The Level Below

OSSEC

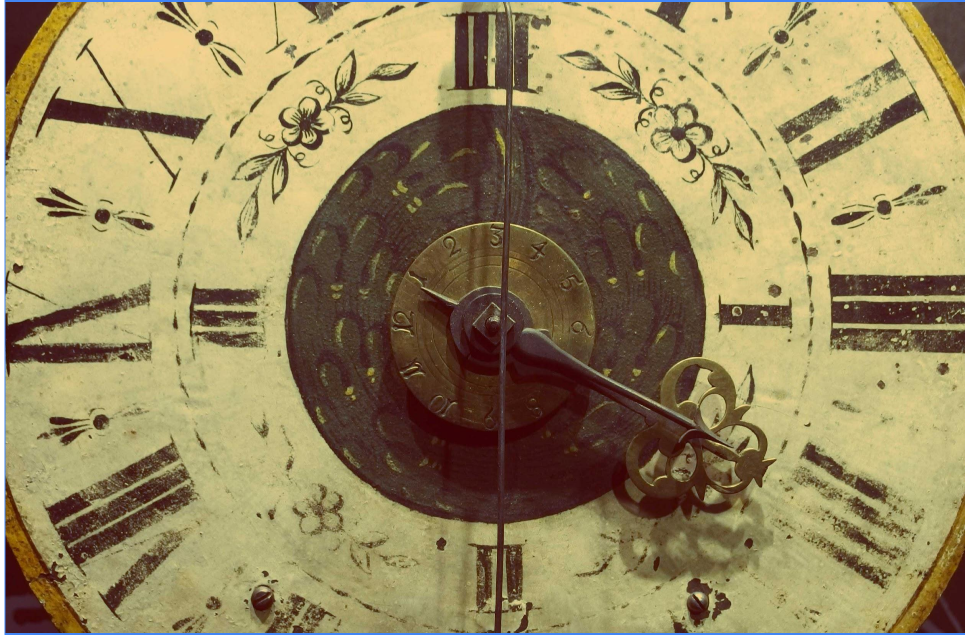
Lynis

Package manager upgrades

Useful Tools Quiz

http://bit.ly/sc2018quiz1

Timing



Halt The World

Stop all feature dev and work only on the upgrade

Works well if code deadlines are sparse or malleable, but is hard on the business

Can bring folks together with a spring cleaning like feel

Doesn’t work on active projects

Little Pieces

For larger upgrades, doing smaller pieces at a time can be useful.

Upgrading to levels where deprications appear then fixing those as dev continues can work as long as the rest of the team doesn’t add more “old style” code

Can take forever to complete, leaving management feeling like it will never finish

Tends to get interrupted by shifting priorities

Iterative Chunks

Do one of the steps in the upgrade dance, then commit that and make it main line code

Each step requires new code to either be tracked and upgraded as it happens or a reconciliation step at the end right before merge.

Basically smaller lock periods than the halt the world approach but without the constant backtracking of each piece

Timing Discussion

Timing

- Halt the World
- Little Pieces
- Iterative Chunks

Approaches

The upgrade dance!

The Step By Step - Upgrade the gem file walk through each step

The Two Step - Upgrade the specs first, then upgrade the gems

The ReRe - Create a clean app and port code over

The Flail and Fail - Things not to do

The Step By Step

Bump major dep versions in the Gemfile

Like Rails version, Hyrax version

bundle update **just those gems**

Look at dependency issues and resolve them, adding gems to your update list until you get a build

Look at https://guides.rubyonrails.org/upgrading_ruby_on_rails.html#the-update-task

The Step By Step

Run rails app:update

Look at release notes from Hyrax

Follow those steps, noting which ones have to be done again on production data when the app is deployed

Run full QA plan... expect issues as there is no test coverage

It can be helpful to get the model layer working first, then go route by route to fix controllers and views*

Exercise - Upgrade a Rails App

Our goal here is to go through the gem process. We’ll do a couple together and then do some in groups.

The Two Step

A lot of the same steps as above, though because you have tests you can have a lot more confidence in the process

Upgrade the specs first, then proceed with upgrading the application

This is the best way to smallish version jumps

The ReRe

Generate a clean app with the latest framework

Copy the specs / spec dependencies over and get them running or upgraded (as in the Two Step)

Copy and update any initializers / config files

Copy and update just the models over and get all model specs passing

Copy any code from lib or other non-standard places

Copy and update assets over, check JS for needed updates

Copy and update controllers and views over one route at a time, modifying as needed

ReRe vs Two Step Exercise

What are some factors between upgrade paths?

Which should we use in these scenarios and why?

The Flail and Fail

The Flail and Fail

Focus, focus, focus

Do the smallest thing you can at a time

Bundle update without arguments is not your friend

Don’t decide to just rewrite the whole app in node/crystal/rust

Minimize refactoring and feature changes during upgrades (see focus above)

Success Metrics - Justify Upgrades

What does success look like?

Code Churn

Total Cost of Ownership

Technical Debt Paydown

Cross Training

Timing & Approach Quiz

http://bit.ly/sc2018q2

And now a word from our sponsor

Questions?

Rob Kaufman
@orangewolf
rob@notch8.com
https://www.notch8.com

This Deck
http://bit.ly/n8sc2018-2

7) Keeping your Samvera application up to date (NOTE: Changed to morning)

Description: This workshop will discuss upgrade strategies, workflows, best practices and common pitfalls. Some familiarity with either Rails or Samvera applications is recommended. We'll go in depth on how we evaluate an application for upgrade, different strategies we recommend and when to use them, and walk through a couple scenarios in depth.

TODO:

Create sample “behind” rails app with bundler audit, retirejs and brakeman

Create quiz for brakeman/bundler audit

Create two step vs rere exercise

Create quiz for timing and approaches?

- When To Upgrade
- How do you measure an upgrade
- Approaches for upgrading
  - The Step - Upgrade the gem file walk through each step
  - The Two Step - Upgrade the specs first, then upgrade the gems
  - The ReRe - Create a clean app and port code over
  - The Flail and Fail - Things not to do
- Timing
  - Halt the World
  - Little Pieces
  - Iterative Chunks
- Success Metrics