Rails Rescue Checklist

http://railsrescuebook.com

Strategic Planning

- [ ] Consult with client on pain points
- [ ] Set expectations
- [ ] Keep track of work to be done

The Opening Investigation

- [ ] Connect to or establish SCM
- [ ] Read any documentation
- [ ] Freeze Rails
- [ ] Install existing database, or run migrations, or load schema
- [ ] Load any seed data
- [ ] Install gems via `rake gem:install` or manually
- [ ] Check plugin README files
- [ ] Survey the models, controllers, views, and helpers
- [ ] Draw or generate an entity-relation diagram
- [ ] Run `rake stats` to see test coverage
- [ ] Determine which testing libraries are in use
- [ ] Read the existing deployment recipe to verify you have the pieces you need

Digging Into Quality

- [ ] Install `metric_fu`
- [ ] Use `rcov` to see test coverage
- [ ] Use `saikuro` to see cyclomatic complexity
- [ ] Use `flay` to locate excess use of cut and paste
- [ ] Use `flog` to see code complexity hot spots
- [ ] Use `reek` to detect common code smells
- [ ] Use `roodi` to see potential errors
- [ ] Read existing logs to find mass-protection errors, hidden errors, and excess SQL
- [ ] Hook up some form of exception notification
- [ ] Run a log file analysis
- [ ] Install `rack::bug`, FiveRuns TuneUp, `rails-footnotes`, or `Palmist` to track activity in the running application

Database Rescues

- [ ] Isolate seed data
- [ ] Refactor procedural migrations
- [ ] Collapse migrations
- [ ] Check for database mismatch

Testing Rescues

- [ ] Check the test coverage
- [ ] Find out if the tests still run
- [ ] Get rid of scaffolded tests
- [ ] Delete completely muddled tests
- [ ] Comment out failing but salvageable tests
- [ ] Choose and install your testing tools
- [ ] Test new code as it's written
- [ ] Add acceptance tests
- [ ] Extend test coverage

Refactoring Rescues

- [ ] Upgrade to current Rails, but cautiously
- [ ] Locate fat controller methods
- [ ] Move excess controller code to models
- [ ] Move to RESTful controllers
- [ ] Extract subsidiary models
- [ ] Move model code to modules
- [ ] Factor out raw SQL in models
- [ ] Remove SQL from views
- [ ] Extract partials from views
- [ ] Extract helpers from views
- [ ] Evaluate and replace plugins as needed

Performance Rescues

- [ ] Test or analyze to locate problems
- [ ] Run Google Page Speed to locate slow pages
- [ ] Implement caching: page, action, fragment
- [ ] Use `memcached` to avoid excess database hits
- [ ] Eliminate N+1 query issues with eager loading
- [ ] Add missing indices