

# The Benefits of a Healthy Rails Application

By Eric Davis | [www.LittleStreamSoftware.com](http://www.LittleStreamSoftware.com)

# TABLE OF CONTENTS

<b>Introduction</b> . . . . .	<b>4</b>
<b>Why application health is important</b> . . . . .	<b>6</b>
1. Software can be an asset for your company. . . . .	6
2. Investing in your software assets can improve their performance . . . . .	7
3. Problem applications can cause negative ROI . . . . .	8
<b>Healthy Rails applications</b> . . . . .	<b>11</b>
1. Is the team making forward progress? . . . . .	11
2. Are investments in the application paying off? . . . . .	13
It's not about the code - at least not directly . . . . .	14
<b>Ongoing process</b> . . . . .	<b>16</b>

# INTRODUCTION

# INTRODUCTION

---

A healthy Rails application is an asset that will benefit your organization for years to come – an unhealthy one, on the other hand, will be a weight you and your team will have to bear every day.

# WHY APPLICATION HEALTH IS IMPORTANT

# WHY APPLICATION HEALTH IS IMPORTANT

---

A Rails application's health is important for a number of reasons.

## **1. Software can be an asset for your company.**

Not in the platitude sense like “our people are our most valuable asset”, but an actual asset: something that helps your business create more money. Something that your business can sell to someone else.

For some companies, their software can even be an asset that makes them unique and stand apart from their competitors. Think about FedEx's routing software and how hard it must work to make their operations function day and night.

Or take something even simpler: my local coffee shop switched to using Square to process credit cards. They didn't need a

# WHY APPLICATION HEALTH IS IMPORTANT

---

## 2. Investing in your software assets can improve their performance

---

custom-programmed cash register anymore, and just used iPads with the Square card reader. It doesn't seem like a huge deal until you compare it to another shop that still does everything manually. This manual shop has to enter prices manually: 50% of the time they charge different amounts for the same drink, or they forget to add an extra charge for a change. Even minor sums like \$0.50 per transaction can add up over time with the volume of transactions done, and cost you dearly.

## **2. Investing in your software assets can improve their performance**

When you have a working asset, you want to continue its cash-flow for as long as possible. Very few assets can keep working forever though: many of them wear out and have ongoing costs, others become obsolete and have to be discontinued.

# WHY APPLICATION HEALTH IS IMPORTANT

---

## 3. Problem applications can cause negative ROI

---

Software can be different.

Software's ongoing costs can be extremely low based on what you get out of it. Once developed and in production, many Rails applications can continue working with a small operating budget for servers and a moderate budget for software maintenance.

Compare that to some machines that need millions spent on maintenance work each year just to continue working. That size of maintenance budget for a healthy software application is unnecessary.

### **3. Problem applications can cause negative ROI**

On the other hand, if you don't have a healthy Rails application, your costs are going to be much higher. An unhealthy software



# WHY APPLICATION HEALTH IS IMPORTANT

---

## 3. Problem applications can cause negative ROI

---

application can eat money faster than an unfed teenager at a buffet.

With software, you're fighting two battles: one to improve the application, and one to repair the application from past decisions.

Making a healthy application from the start will need less repairing, and your investments can go towards improvements instead. An unhealthy one will need a lot of repair work – sometimes so much that it doesn't move forward at all.

# HEALTHY RAILS APPLICATIONS

# HEALTHY RAILS APPLICATIONS

---

But what makes a healthy Rails application?

There are two overarching criteria to use to determine if an application is healthy or not.

1. Is the team making forward progress?
2. Are investments in the application paying off (positive ROI)?

Though this may sound simple, there is a surprising amount of depth to these two points.

## 1. Is the team making forward progress?

Forward progress means anything that makes the application **better** over time. While this is subjective, every application's charter will be organized around something that describes how

# HEALTHY RAILS APPLICATIONS

---

## 1. Is the team making forward progress?

---

its progress should be measured. Even vague ones like “Empower airline fliers to find the best deals” can provide guidance.

This can also be referred to as “the problem this software solves”.

Forward progress may be how quickly new features are added to a startup’s SaaS product, how quickly maintenance requests are fulfilled for a back-office application, or the total cost savings from a business process automation system.

The key to forward progress is that things improve. Better. More. Faster. Less. Cheaper. Choose whatever descriptions fit, but choose something your team can agree on.

# HEALTHY RAILS APPLICATIONS

---

## 2. Are investments in the application paying off?

---

### **2. Are investments in the application paying off?**

The second criteria is that investments on the application are paying off. This is directly related to the first point, but on a financial basis.

Does the new \$5,000 server process the nightly batch operation in 10% of the time?

Is the new developer generating 200% of her salary in value?

Will the optimizations pay for themselves over the next 5 months?

Many of the discussions around investments will put a measurement on top of forward progress. “Will X in Y?” “Pay \$5,000 now to save \$20,000 later”

# HEALTHY RAILS APPLICATIONS

---

It's not about the code – at least not directly

---

## **It's not about the code – at least not directly**

If both of these criteria are met, the application is deemed healthy.

That means that in theory, your Rails application can have horrible code, breaking every “community quality standard”, and still be considered healthy – as long as you have the best team and best environment.

But stop and think about how much better application's return could be if it had good, or better, great code!

The quality of the code affects the team and how work is done in the application. This affects how difficult forward progress is, which then affects any ROI.

# ONGOING PROCESS

# ONGOING PROCESS

---

A Rails application needs maintenance and repair to keep working, much like any other asset. However, it can also be optimized to achieve an even greater return.

Before you can enhance the return, you need to make sure your application's health is high, and that it stays that way. Make sure the application's life is as long as possible and you don't lose the current return.

If you'd like to talk to me about your Rails application's health, I'd be happy to provide a free consultation to go through my multi-part evaluation process. To start, [go here](#) and enter your name, email address, and a short description of your application. I'll then contact you for more information.

To your application's health,

Eric Davis, Little Stream Software