reinteractive

Our Clients Trust Us

QANTAS

AUSTRALIA POST

Stockland

STATE LIBRARY®
NEW SOUTH WALES

Penfolds®

ADELAIDE FRINGE

Westfield

amaysim

THE UNIVERSITY OF SYDNEY

ABC Commercial

nab

+ over a hundred more…

reinteractive | CONFIDENT SOFTWARE

We are a web development and design team that **puts communication first.**

We've developed software used in more than 2 million applications world wide.

Over 200 applications built; ranging from enterprise applications to startups.

Our Expertise

Large scale complex applications that work with a variety of technologies.

From project conception and design, through to market deployment, hosting and ongoing support.

salesforce registered consulting partner

aws partner network | Advanced Consulting Partner

> *The expertise, the direct communication with the developer and the seamless project management has been fantastic. Although I wasn't familiar with Ruby on Rails, reinteractive really put me at ease.*
>
> Jenna Bain, State Library of NSW

> *We engaged reinteractive because of our shared understanding of iterative, agile development. Initially we worked with their team on discrete development projects, but over time our relationship has grown, such that they attend our daily stand-ups and have become a part of our collaborative cross-functional teams.*
>
> Ian Tinsley, Westfield Labs

> *The difference with reinteractive is the talent pool – they have captured the best talent in the country. The consultants are top resources; self-motivated, talented, passionate and up front.*
>
> Simon Farrel, redballoon

Mikel Lindsaar

Founder & CEO

reinteractive was founded by Mikel in 2010 and has quickly grown to become the largest Ruby on Rails focused development firm in Australia.

He is a leader in the open-source field, having re-written the Ruby on Rails mail handling library, which has now been downloaded over 110 million times.

Mikel is a member of the Rails commit team, an international team of peer-approved individuals who are authorised to make or approve changes to the core Ruby on Rails code base.

Mikel's software is used by over 2 million applications worldwide including:

Bloomberg airbnb fiverr® Basecamp® GROUPON shopify twitter

App **Review**

for **Ruby on Rails** Applications

App Review

Is the App Review service right for you?

The **reinteractive App Review** service is the industry leading analysis service for **Ruby on Rails** applications. It is a 7 point review of your application to provide you with essential information around security, performance, risk management and a summary of quality and technical debt within your application.

The review is conducted by our **Quality Division** team members who are the highest trained, most knowledgable and attentive to detail, so you are assured an accurate review.

Who Is It For?

What is our App Review service?

The **App Review** is highly valuable if you are in any of the following situations:

1. You have a Rails application with no or only a few developers working on it

2. Development was completed by another software firm and you need to verify the quality of the app

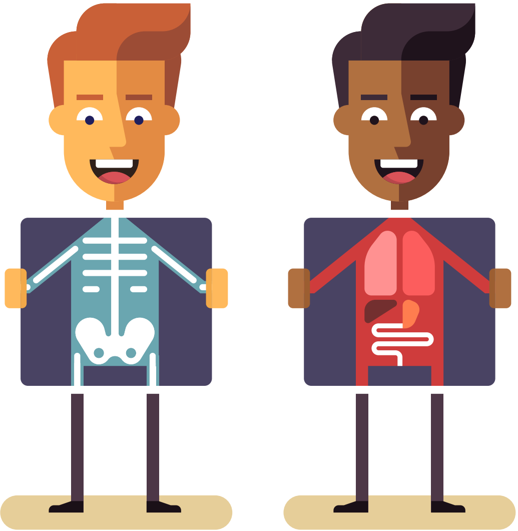
3. You are experiencing performance issues

4. You need to determine the risk profile of your investment

5. It is important to know if you have any security issues within your application

6. You want to develop new features and need a clear picture of your app as a base line

7. You want a sense of the technical debt within your application and what may be required to clean it up



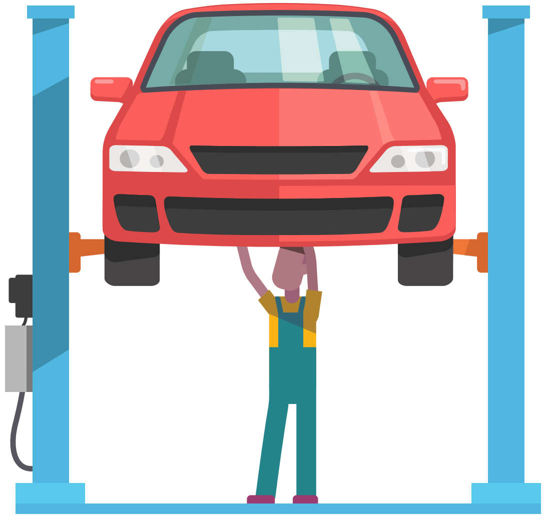
Why Perform an App Review?

Should you get a Review on your Rails App?

You have invested a lot of money in your application, but do you have a complete understanding of the risks within your code?

If you were looking to buy a car you would need to be aware of any risks. You would always engage a mechanic to inspect the foundation, structural integrity, electrical wiring, etc.

In the same way before engaging in further development, understanding your application’s risks will save you a lot of money and development time in the future.



App **Review**

the **Elements** of the Review

Elements of the Review

Setup & Documentation

How well documented is your app. How easily can a new developer get up to speed.

Automated Tests

How well your application is covered by automated tests can determine the cost of future development.

Database Setup

Are your databases setup correctly or do you have sensitive data stored where a hacker can find them.

Elements of the Review

Performance

Do you have any performance bottle necks that could be driving people away from your site.

Security

Is your client or sensitive information secure. Are there any known security vulnerabilities within your application?

Versions

Are there any known upgrade issues within your code. Knowing these can make version upgrades easier.

Elements of the Review

Code Quality

How well has your code been written. Good quality code equals maintainability and cheaper future development.

Sensitive Data

Ensuring sensitive data is stored following best practices is essential in keeping your app secure.

Technical Debt

Poorly written code can result in bugs, performance issues and security concerns.

Step 1: **Setup**

Can a new developer get up to speed quickly?

How well is your application documented?

This is important, as each new developer comes to work on the application, how quickly they can get up and running as this can save you time and money.

A well written and documented Rails application should be able to get setup and running on a developer's computer in around 15 minutes.

The review will assess how long this takes and how well documented the application is for a new developer.

Getting this right results in an increased speed of development.

Step 2: **Automated Testing**

Is your application designed for future maintainability?

Test driven development is part of a well designed and written application.

Test driven development (TDD) is a software approach in which a test is written before writing the code. That code is developed until it passes that test.

With internal tests written into your application, as new code is developed it is run against the pre-existing tests. This will help to determine whether this new code will break anything that is already written or not.

This stage of the review will determine whether the existing tests pass and what percentage of the application is covered by tests.

Ensuring your test coverage is at a decent level and all tests are passing, provides confidence in the ability to extend the features of the application.

Step 3: **Database**

Is your database setup correctly?

This stage of the review will assess the structure of your application’s databases to ensure they are setup correctly for data security and potential security risks.

Does the application store credit cards in the database? This would be a big red flag.

Is sensitive information encrypted to ensure privacy of your customers information.

This and many other points are assessed against your database.

Step 4: **Performance**

Does your application have performance bottle necks?

In today’s consumer environment, most people expect pages to load within 2 - 3 seconds.

Did you know site speed is a factor in Google rankings.

Loading time is a major contributing factor to page abandonment. 25% of users will abandon your site if it takes four seconds to load, and it get worse from there.

The App Review will look for and note any performance bottle necks within your code.

Step 5: **Security Analysis**

Are there any know security risks in your application?

Nearly ever week we hear about a major web company who experience a major data breach.

The review will analyse your application to determine any known security flaws that could be exposed by hackers and therefore put your client’s sensitive data at risk.

The review will sniff out known issues so you have the knowledge to do something about it.

Step 6: **Versions**

What are the current versions of Ruby, Rails and your code libraries?

Maintaining Ruby, Rails and your associated code libraries (gems) at their highest level is a required part of looking after your application. As your car requires regular maintenance, so does your application.

The longer you leave it to upgrade your code, the more expensive it costs. Furthermore the longer you leave it the more difficult it becomes as certain gems become redundant or not supported by future versions.

We will determine your current versions and if there are any known upgrade issues so you can begin to plan out your maintenance road-map.

Step 7: **Code Quality**

Is your code written for maintainability and new features?

There are two major ways to write code.

1. Code right the first time with emphasis on easy to read and understand code so the next person working on it can continue without pause.

2. Writing spaghetti code, enough for the application to work, but it makes it very difficult for new developers to add features without spending a lot of time understanding the code.

Point 1 appears more expensive in the short term, however, in the long term bad code has an exponential development cost, sometimes to the point where it is cheaper to rebuild from scratch.

The review with provide you with a summary of the quality of your code.

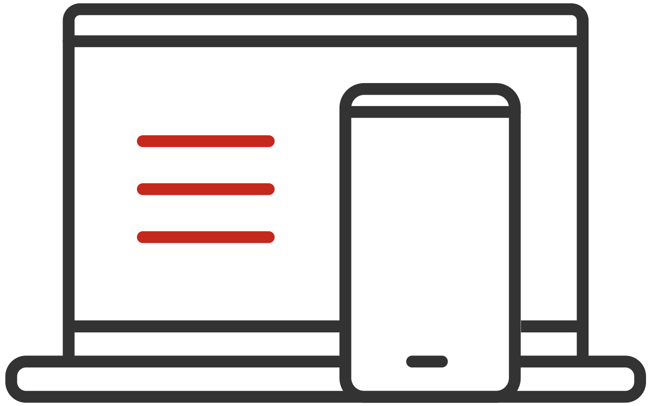
Step 8: **Sensitive Data**

Is sensitive data stored in the correct areas?

The storage of sensitive information is a big issue in development. Right now there are hundreds of bots that scan through online code repositories looking for access codes to things like accounting software, AWS, and other important components of your application.

Many times we have discovered sensitive credentials stored in the code of an application making it more easily accessible to hackers.

Ensuring sensitive data is stored following best practices is essential in keeping your app secure.



Step 9: **Technical Debt**

Are there code issues that require rework?

Technical debt is a concept in programming that reflects the extra development work that arises when code that is easy to implement in the short term is used instead of applying the best overall solution.

Technical debt is an outgrowth of code quality. Every shortcut in development will almost inevitably need to be fix up at some future data, otherwise development costs will continue to increase.

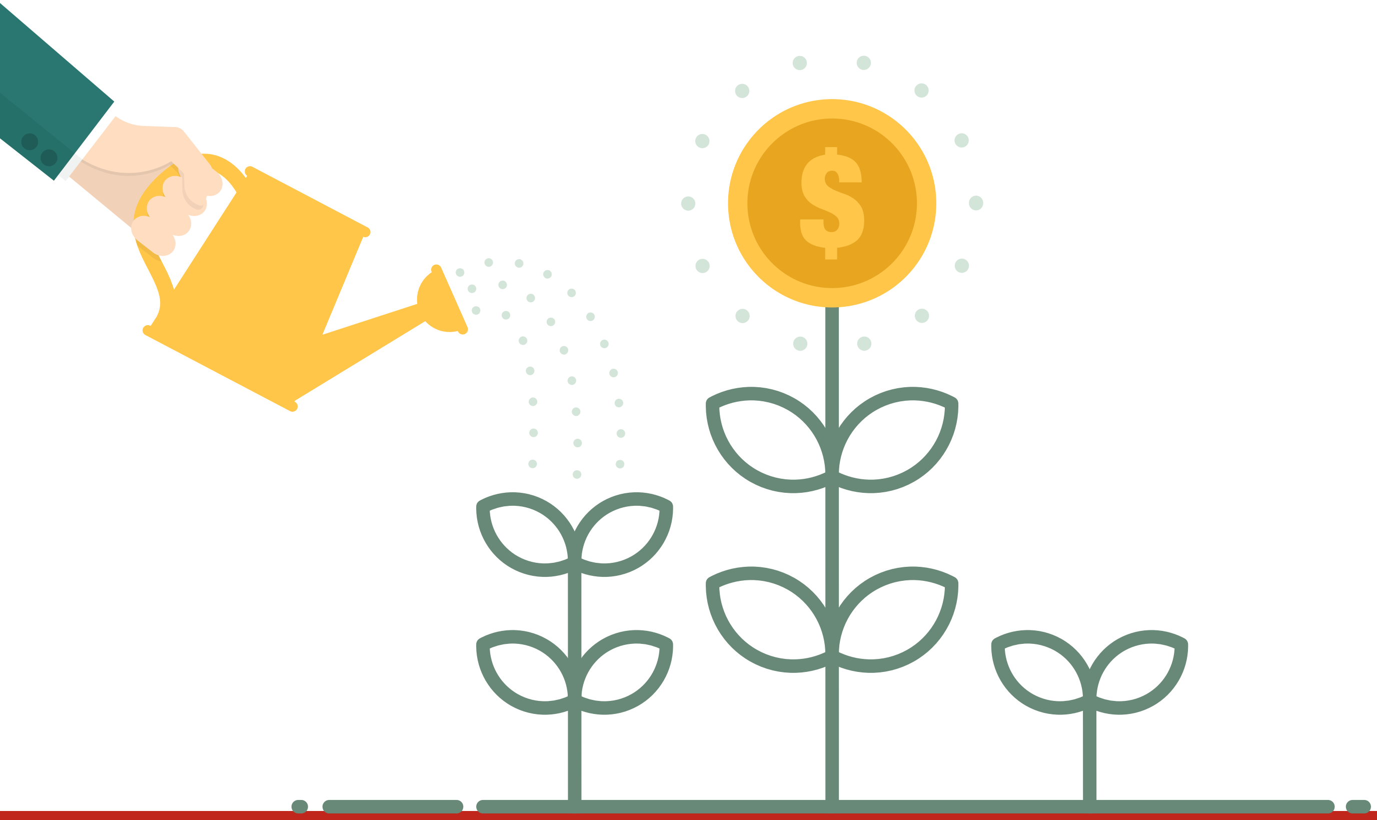
Code that is well written from the start will have very little technical debt.

Poor code quality will need its technical debt ‘paid down’ at some point.

The review will give you a summary of the quantity of technical debt that you have in your application.

Your **Investment**

How much does the inspection cost?



The cost of the **App Review**

$950 + GST per application

Time

How long does the App Review take?

Once you have signed up for the **App Review** it will take approximately 3 - 5 days to deliver it.

Final Result

What do you receive at the end of the inspection?

At the completion of the App Review you will receive a full report and analysis that:

1. Detailed findings of all key areas of the application, the good and the bad

2. Recommendations on how to improve the application listed by priority

3. A review call with a Rails developer on these findings so you fully understand the state of your application

salesforce registered consulting partner

aws partner network | Advanced Consulting Partner

Let’s get Started

reinteractive

AUS +61 2 8019 7252 USA +1 415 365 9680 info@reinteractive.net