The Craft of Developing Ruby Applications

Overview

Ruby application development is a craft just like other professions that requires years of practice and experience. An expert in the craft is a lot different than a programmer who is only able to write Ruby code. Most programmers undergo a transformation in their careers where they learn they have been writing bad code in one form or another and seek out different ways to improve their skills.

One of the most accessible sources for information and knowledge on the subject of application development are books. A fair number of books on Ruby have already been written that cover the lexical and syntactical structure of the language. Very few books have been written on the craft of application development with none of them covering the Ruby programming language. This leaves readers without a simple source where they can learn about their craft while at the same time improving their command of the language.

The craft of application development encompasses many things. It requires the practitioner have a solid command of the language, understanding its nuances, working around its limitations while using its features to improve productivity. Tools are an essential part of daily exercise of writing code. Depending on the project one may require the ability to use a profiler or acquire a deeper understanding of a libraries' internals by stepping through the code with a debugger. Lastly there is a required understanding of the Agile Methodology that guides the processes around the code development.

The purpose of the book in this proposal is to provide a simple guide to Ruby application development that educates the reader on the craft and not just the code. It will introduce the reader to requirements for a reporting framework that needs to be developed through User Stories. The reader will be shown how the requirements turn into unit tests that lay the groundwork for Test Driven Development using RSpec. The book will simulate the real world changing of events that happen during the course of application development. Changing requirements will be used to show how unit tests provide quality assurance when code is rewritten or refactored. Built in Ruby design patterns will be introduced to demonstrate how to mitigate changes that occur during the development life cycle. A style guide will be discussed as a way of assuring the code is accessible to other developers and testers. Social coding will be covered to illustrate how to take advantage of other peoples' knowledge of a particular domain to improve the code base. Upon completion the narrative in the book will have covered the tools and techniques that are used by Ruby experts.

There are two things the author believes are an important part of this proposal. The first is that I love being a practitioner of Ruby application development and teaching others the craft that puts them on the path of becoming an expert in the field. I also have love for books, specifically computer books that cover just about any topic on programming languages. The lack of a book that provides a simple read on becoming a better ruby programming is motivation for the author to write about the knowledge he has acquired over the years. It would be a source of pride to be able to point people to a book where they can actually learn there is more about Ruby than learning how to write a class or perform a conditional comparison. Let's take the opportunity to write a book that fills the gap between syntax and the techniques that are required of real professionals every day.

Marketing

Promotion

Competing Books

As mentioned in the Overview there are a wide variety of books written about the Ruby programming language. Quite a few from various publishers cover the lexical and syntactical details of the language for intermediate and advanced programmers. Other books serve as an introduction to the basic concepts of the language and its usage.

The other type of book that is of interest are the ones that cover software development methodologies. These books are language agnostic and present the principles and guidelines for writing great software with the focus on the events and routines that make teams highly productive. This type of book rarely has any code examples to illustrate the concepts that are covered.

The author has identified several books that compete in the market for the proposed book, each listed below with their current Amazon best seller rank:

- Clean Code: A Handbook of Agile Software Craftsmanship
- The Passionate Programmer
- The Pragmatic Programmer
- Test Driven Development: By Example
- Code Complete: A Practical Handbook of Software Construction
- Ruby Best Practices
- The Ruby Programming Language
- Programming Ruby 1.9: The Pragmatic Programmers' Guide
- Beginning Ruby: From Novice to Professional
- The Ruby Way, Second Edition: Solutions and Techniques in Ruby Programming
- Learning Ruby

About the Author

The author has been developing applications since he was 12 years old and wrote a program in Basic on a TRS-80 Color Computer II. He has over 15 years of professional experience developing applications in a variety of languages but started writing Ruby in 2007 and has never looked back. He is an active contributor to the ruby message boards, has written a few useful gems and has contributed patches to the Ruby on

Rails core. The author loves to teach and mentor other professionals so that they get better at writing application code in Ruby with a strong foundation in Object Oriented techniques and Agile development methodologies.

List of Chapters and Summaries

- *Introduction*

  Expand on the premise of the book about it being a tool to learn about the techniques and tools that take a person from being a programmer to a practitioner of the craft.

  - intended audience
  - how to use the book

- *The User Story*

  Every application that is developed must have requirements regardless of the complexity. The best way to document those requirements are by writing about the user's experience with it. User stories can be daunting for beginners because they require experience in what it means to write a good one. This chapter will discuss how to write user stories and what role they play in the development process.

  - What is a user story?
    - who are the users
    - define the system boundary
    - define what the user wants to accomplish
  - who writes a story
    - Solo
    - product
    - team
  - sizing a story
    - relative scale
    - who sizes a story

        - relative difficulty
        - planning poker
    - the right size
        - not enough detail
        - too much detail
    - Takeaways
        - recognizing epic stories
        - not a user goal
    - tasking a user story

- *Behavior and Test Driven Development*
- *Continuous Integration*
- *Social Coding*
- *Style Guides*
- *Refactoring*
- *Writing a Gem*
- *Meta-Programming*
- *Profiling/Benchmarking*
- *Static Code Analysis*

Sample Chapter