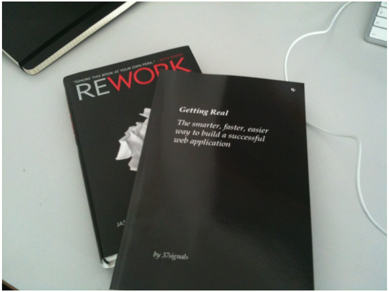
Getting Real – The Secrets of 37signals

George Rogers

March 25, 2011


"IGNORE THIS BOOK AT YOUR OWN PERIL."
REWORK
Getting Real
The smarter, faster, easier way to build a successful web application
by 37signals

What You Will Learn

What You Will Learn

- About 37signals.

What You Will Learn

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests

What You Will Learn

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests
- Process
  - Race to running software
  - Iterate
  - Done!

What You Will Learn

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests
- Process
  - Race to running software
  - Iterate
  - Done!
- Code
  - Less Software
  - Optimize For Programmer Happiness
  - Refactor, Refactor, Refactor!

WHY?

WHY?

- 37signals is a profitable company; that produces products: on time, and on budget

WHY?

- 37signals is a profitable company; that produces products: on time, and on budget
- 37signals makes some of the most useful web applications with a small team.

WHY?

- 37signals is a profitable company; that produces products: on time, and on budget
- 37signals makes some of the most useful web applications with a small team.
- They made the best web framework: Ruby On Rails.

About 37signals

Products

About 37signals

Products

- Basecamp – A Project Manager and Their First Product.

About 37signals

Products

- Basecamp – A Project Manager and Their First Product.
- Highrise – A Contact Manager

About 37signals

Products

- Basecamp – A Project Manager and Their First Product.
- Highrise – A Contact Manager
- Campfire – A Corporate Chat Service

Feature Selection

Less Is More

Less Is More

- Less Features = Easier To Use

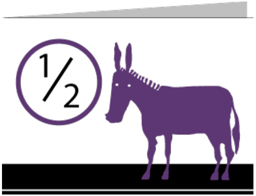
Less Is More

- Less Features = Easier To Use
- Some Features ARE Misfeatures

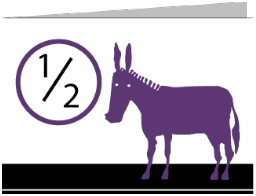
Less Is More

- Less Features = Easier To Use
- Some Features ARE Misfeatures
- Less Features = Less Work

Half NOT Half Hearted

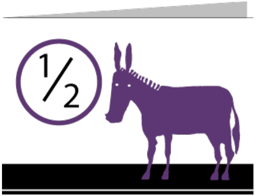


Half NOT Half Hearted



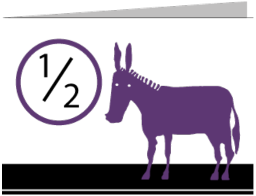
- It is better to have HALF of the ideal product, than have a slipshod implementation of the whole shebang.

Half NOT Half Hearted



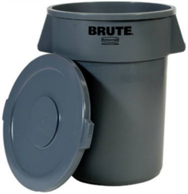
- It is better to have HALF of the ideal product, than have a slipshod implementation of the whole shebang.
- Have you ever used something where the features get in the way. Those features are called Misfeatures.

Half NOT Half Hearted

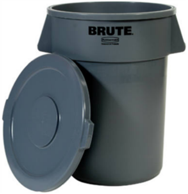


- It is better to have HALF of the ideal product, than have a slipshod implementation of the whole shebang.
- Have you ever used something where the features get in the way. Those features are called Misfeatures.
- Your product will be more focused.

How to process feature requests



How to process feature requests



1. Receive it

How to process feature requests

1. Receive it
2. Read it

How to process feature requests

1. Receive it
2. Read it
3. Forget it

How to process feature requests

1. Receive it
2. Read it
3. Forget it

The good feature requests: you will remember them because you receive them very often.

Process

Race To Running Software

Race To Running Software

- Prototype DON’T plan

Race To Running Software

- Prototype DON'T plan
- Prototypes will help

Race To Running Software

- Prototype DON'T plan
- Prototypes will help
    - Increase agreement

Race To Running Software

- Prototype DON'T plan
- Prototypes will help
  - Increase agreement
  - Improve the quality of your product

Race To Running Software

- Prototype DON'T plan
- Prototypes will help
  - Increase agreement
  - Improve the quality of your product
  - Select the right features

Iterate

Iterate

- Iteration allows you to make decisions by trying things out

Iterate

- Iteration allows you to make decisions by trying things out
- If they don't work just throw them out

Iterate

- Iteration allows you to make decisions by trying things out
- If they don't work just throw them out
- Reduces cruft by figuring out whats really necessary.

Done!

Done!

- Reduce scope to meet deadline – not quality or cost

Done!

- Reduce scope to meet deadline – not quality or cost
- Sacrifice Features – Not Quality

Done!

- Reduce scope to meet deadline – not quality or cost
- Sacrifice Features – Not Quality
- Don’t throw programmers at the problem

Code

Less Software

- Less Software = Better Software

Less Software

- Less Software = Better Software
- Less Software Costs Less

Less Software

- Less Software = Better Software
- Less Software Costs Less
- Less Software = Less Bugs

Optimize for Programmer Happiness

Optimize for Programmer Happiness

- Programmers are people

Optimize for Programmer Happiness

- Programmers are people
- People like doing more work with less effort

Optimize for Programmer Happiness

- Programmers are people
- People like doing more work with less effort
- Select programming languages that make programmers happy

How?

Print to the console the numbers 0 through 9.

Ruby

```
for i in (0..9)
  puts i
end
```

Java

```
class Hello {
  public static void main(String[] args){
    for(int i=0;i<10;i++){
      System.out.println(i)
    }
  }
}
```

Refactor, Refactor, Refactor!

- Refactoring is the process of cleaning up code

Refactor, Refactor, Refactor!

- Refactoring is the process of cleaning up code
- Code can be written in a way that is messy

Refactor, Refactor, Refactor!

- Refactoring is the process of cleaning up code
- Code can be written in a way that is messy
- If you don’t clean it up it will get messier and messier

What You Learned

What You Learned

- About 37signals.

What You Learned

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests

What You Learned

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests
- Process
  - Race to running software
  - Iterate
  - Done!

What You Learned

- About 37signals.
- Feature Selection
  - Less Is More
  - Half NOT Half Hearted
  - Forget Feature Requests
- Process
  - Race to running software
  - Iterate
  - Done!
- Code
  - Less Software
  - Optimize For Programmer Happiness
  - Refactor, Refactor, Refactor!