Introduction

“Engineering is easy. People are hard.”

Like most computer geeks, we discovered that our hobby and passion -- playing with computers -- was a great way to make a living after graduating college.

Trying to write great software? This is for you.

The Myth of the Genius Programmer

The Genius Myth

Deep down we all secretly wish to be geniuses. The ultimate geek fantasy is to be struck by an awesome new concept. But hold on: let’s do a reality check. You’re probably not a genius.

No offense, of course -- we’re sure you’re a very intelligent guy or gal. But do you realize how rare actual geniuses really are? Sure, you write code, and that’s a tricky skill that probably puts you in a bracket above a lot of the human population. But even if you are a genius, it turns out that that’s not enough. Geniuses still make mistakes, and having brilliant ideas and elite programming skills doesn’t guarantee that your software will be a hit. What’s going to make or break your career is how well you collaborate with others.

Hiding is Considered Harmful

If you spend all your time working alone, you’re increasing the risk of failure and cheating your potential for growth.

Bus factor (noun): the number of people that need to get hit by a bus before your project is completely doomed.

It’s All About the Team

Let’s put the idea into simpler words: Software development is a team sport.

The Three Pillars

Almost every social conflict can ultimately be traced back to a lack of humility, respect, or trust.

HRT in Practice

Lose the Ego

Learn to Both Deal Out and Handle Criticism

Your self-worth shouldn’t be connected to the code you write. To repeat ourselves: you are not your code.

Fail Fast; Learn; Iterate

Leave Time for Learning

Learn Patience

Next Steps

Building an Awesome Team Culture

What Is Culture?

Why Should You Care?

If you don;t put effort into building and maintaining your culture, your team will eventually be overtaken by a strong personality who cultivates his culture in your team.

Culture and People

没看懂。

Communication Patterns of Successful Cultures

If you examine any successful, efficient engineering culture, you’ll find high value placed on numerous channels of communication, such as mailing, lists, design docs, mission statements, code comments, production how-tos, and more.

A good general rule around communication is to include as few people as necessary in synchronous communication (like meetings), and to go for a broader audience in asynchronous communication (like email).

High-Level Synchronization

The Mission Statement -- No, Really

A mission statement helps your team confront differences and come to an agreement. ...?

Efficient Meetings

Working in a “Geographically Challenged” Team

Design Docs

Day-to-Day Discussions

Mailling Lists

Online Chat

Using an Issue Tracker

Communication as Part of Engineering

Code Comments

Putting Your Name in Source Code Files

Require Code Reviews for Every Commit

Have Real Test and release Processes

It Really Is About the Code After All

There is first chapter content in the book. Because the book is about team culture, I lost interest in it.