

Learning and Maintaining Code



When You Are The New Guy or Preparing For One

Congratulations!



Jobs and Roles Change

**Unfortunately, that often means
someone has to handoff and/or learn
new code.**

First Steps

- ✦ Think of Learning a New Job as a Journey
- ✦ The first step is to determine where you are
- ✦ The next step is to figure out where you want to go
- ✦ Next you fill in the steps between

Determine Where You Are

- ✦ Is Source Code Available?
- ✦ Is Documentation Available?
- ✦ Any User Manuals, Guides, or Requirements?
- ✦ How About Training?
- ✦ What are the primary technologies and platform?

Still Getting Bearings...

- ◆ Who are the primary users?
- ◆ Is this in production?
- ◆ What is the general impression of the application?
- ◆ What are the strengths and weaknesses of the application?
- ◆ What problem does this solve?

Still Being Still...

- ✦ Gather emails and phone numbers
- ✦ Chat and related sessions
- ✦ Processes and Procedures
- ✦ Coding Standards

The Technical Work

- ✦ Checkout source code
- ✦ Build a development environment
- ✦ Run unit tests if they exist and review
- ✦ Build from version control
- ✦ Smoke test your app

Your Destination

- ✦ Clarify who is on the team
- ✦ Learn/Confirm roles and expectations
- ✦ Who are the support staff?
- ✦ Who are the stakeholders?
- ✦ Are SMEs available?

Finding The Goal

- ✦ What is the expected schedule?
- ✦ Agile? Release Schedule?
- ✦ Find and resolve issues
- ✦ Reporting and defining “done”

The Journey Beings

- ✦ Start simple, build, small change
- ✦ Test your debugger
- ✦ Test, Commit, Review Cycle
- ✦ Communicate Early and Often
- ✦ Do not be afraid of Ignorance
- ✦ No Assumptions

The Daily Grind

- ✦ Regular checkins with business users
- ✦ Understand the industry
- ✦ Magazines, Podcasts, and Blogs
- ✦ Utilize Training and Conferences
- ✦ Look for improvements in daily work

Slow And Steady Wins

- ✦ Nail the Processes
- ✦ Master The Business
- ✦ Master The Environment
- ✦ Know the System, not just your module
- ✦ Underpromise and Overdeliver

Ramping Up is Expected

Make The Team Better

Be A Team Player

Passing It On

- ✦ Embrace Standards and Underpromise and Overdeliver
- ✦ Documentation and Comment Code
- ✦ Test and Build Scripts
- ✦ Automate Where Possible
- ✦ Avoid Technology Scatter

Thank You!

I appreciate your time.

Please send any questions, comments, or requests for assistance to info@develpreneur.com or contact us on the site.

Our goal is making better developers everywhere and at all levels.