Discipline and Professionalism – Are you a Software Craftsman?

ontheagilepath.net/2015/06/discipline-and-professionalism-are-you-a-software-craftsman.html

By Sebastian Radics

This week I attended the great Software Craftsmanship Berlin Meetup to listen to Daniel Irvine's great talk about Discipline and Professionalism in Software Development.

I was deeply impressed by the high value attitude and with this post I share my notes and insights from the talk.

A short summary:

Discipline	Professionalism
Little things to improve skill	Trust from others
Refactoring	Discipline is essential
Naming	Code ownership
Practice with katas	Shipping with no known bugs
Good commits	Being responsible
The zone	Respecting others
Pairing	Working with change
More discipline = more skill	

THE CRAFTSMANSHIP TEST

Read on to get some more details about discipline, professionalism and the Craftsmanship Test.

Discipline

The following highlights some important aspects of discipline as a software developer.
Craftsman cherish CODE QUALITY. They are pride of their creations and create to the best of their abilities.
Quality is the most important thing.
Refactor mercilessly – as you will very likely not return to your Todos.
Refactor systematically and in small steps.
Commit often and don't brake the code while refactoring and committing (even if it implies some more steps in between).
Use "good" commits – using descriptive commit messages, creating working software with each commit, each commit does one thing and have many commits during a day.
Practice, **PRACTICE**, PrAcTiCe – best every day, e.g. in the morning using Coding Katas to engage your brain and extends your knowledge every time.
Prefer pair programming as the engaging variant over the Zone and autopilot brain mode. It's difficult but keeps you engaged.
Writing prose – e.g. by writing a blog. It enables deeper learning and connects hunches.
Follow e.g. John Sonmez – an awesome blogger with a great course for developers that want to learn blogging.
By writing regularly you discipline yourself a lot.

Being disciplined is the requirement for being professional.

Professionalism

What are cornerstones of being a professional software developer?
Create trust – by connecting and working with your stakeholders. Keep in mind that we trust those who act like us … and often building a bridge is a key success factor for creating trust.
Micromanagement is a clear sign of missing trust.
Avoid a not my code attitude.
Apply a ZERO bug policy. No bug backlogs! Shipping code with known bugs is crazy.
Teach – your junior comrades.
Respect – build on a daily base, have fun, do silly things together.
Make decisions and don't avoid it – e.g. for Libraries, Tools, Coding standards and necessary changes.
Engage in Life Long Learning.

The Craftsmanship Test

In addition to the great Joel Test (12 steps to better code) and John Sonmez great extension – The simple programmer test – Daniel proposed the following questions for the Craftsmanship Test:
Is pair programming encouraged?

Do you make continuous process improvements?

Do programmers take responsibility?

Do you ship with zero bugs?

Do developers get time to practice?

Do developers commit to life long learning?

Book recommendations

Refactoring by Martin Fowler, Kent Beck, John Brant, William Obdyke, Don Roberts
Implementation patterns by Kent Beck
Clean Code by Robert C. Martin
The Clean Coder by Robert C. Martin
Mastery – The Keys to Success and Long term Fulfillment by George Leonard
Software Craftsmanship – The new imperative by Pete McBreen
CRAFT = ART + SKILL
Have fun on your journey! Maybe you can add even more hints? Please share via your comment.