

The Software Craftsman: Professionalism, Pragmatism, Pride

Written by Sandro Mancuso, and published by Pearson/Prentice Hall Professional, © 2014, paperback, 9780134052502, pp.288, \$30.

Reviewed by Ryan Dawson.

The Software Craftsman is a handbook in how to follow the values of enlightened professionalism that the growing craftsmanship movement celebrates. The unifying aim of craftsmanship is to improve the IT industry by inspiring software engineers to act with greater autonomy, collaboration and integrity. The model craftsman cares about the value of their work as a whole in a way that goes against the grain of traditional segregations of roles in IT projects. The craftsman is not only concerned with what they personally produce but also with seeing that what they deliver is of value and part of an overall success. This means a strong emphasis on clarity of the product's purpose and the quality of its code. It also means that the model craftsman takes on responsibility and is prepared, where necessary, to be involved in any aspect of the delivery process. The book's anecdotes illustrate clearly some of the industry's present problems and how these problems are connected with traditional attitudes to software roles and to software quality. The book offers strategies and inspiration for how to address these problems by adopting the values of craftsmanship in working life.

Mancuso explains the present phase of growth of the craftsmanship movement as a reaction to the 'Agile hangover' - the realization that the adoption of Agile project management practices alone has not solved all of the industry's problems. The craftsmanship response is to both emphasize the less widely-adopted technical practices associated with agile (such as pair programming and test-driven development) and to suggest the adoption of a particular set of values, attitudes and ways of working. The values and approaches are brought out through anecdotes illustrating Mancuso's own rocky journey from naive young developer to experienced craftsman. The reader is shown that progression along the craftsmanship journey is not just about acquisition of technical knowledge and skills and is crucially about maturity of approach.

The theme of maturity is best illustrated in the book's discussion of how to deal with pressures arising in projects that are in deep trouble. Many developers have had the experience of being told that they must deliver something in a timeframe that is unrealistic or of being asked to build a solution from requirements which don't make sense. *The Software Craftsman* advises developers to accept that these situations will arise and learn how to deal with them calmly and professionally. It emphasizes that the 'hero' approach of coding day and night is rarely the best one and may not be a solution at all. The book illustrates how to deliver a positive message along with the negative - how to focus on what can realistically be done and how to move forward. The book is sensitive to the subtlety of such situations and how difficult it can be to calmly explain what is realistically achievable when under this kind of pressure.

The Software Craftsman is broad in its reach, covering many different aspects of the software landscape as it sets out its vision for a better software community. Running throughout the discussions are themes of pragmatism and mutual respect and these themes are especially prominent in the discussion of recruitment. The book criticizes practices of putting candidates through tests of obscure knowledge or unusual puzzle-solving. It emphasizes the straightforward message that the best interview is being able to sit down and work with the candidate and the next best is being able to have an enlightened technical conversation. It can be tempting to undervalue the importance of free conversation to the interview process as decisions based upon conversations can seem subjective. However it is important to be clear that it is normally sound judgment that we should be looking for in software engineers. There is no better way to get a sense of somebody's judgment than understanding the way in which they face relevant situations and the way in which they think.

The book discusses the importance of the technical practices of agile and argues that a more wholehearted adoption of practices such as test-driven development and pair programming would lead to a greater success rate on Agile projects. But *The Software Craftsman* is not a book about these practices and it does not contend that the blind adoption of these practices alone would solve all problems. Mancuso sounds notes of caution against writing automated tests for the sake of having tests or trying to test things for which automated testing gives little benefit. For Mancuso these technical practices have most importance in the way in which they resonate with and promote the values of craftsmanship.

In a sense *The Software Craftsman* is an advice book - it gives specific ways of approaching development and being a developer that aim to make developers better. It is also an expression of how to be part of the craftsmanship movement and what that movement means. As an advice book it strikes a personal tone and its advice is well-observed and often deeply relatable. It can sometimes come across as opinionated and one should perhaps expect a book to be opinionated which puts forward a vision for an alternative set of values. I suspect that most readers will not agree with every opinion expressed and I don't think it is the book's intention that they should. For Mancuso craftsmanship is more a set of attitudes than a creed and different people can be craftsman in different ways.

A shift towards craftsmanship may not be an easy one given that some of the industry's practices tend to run against it. The honesty and forthright nature of the book could help give developers courage to try to do things better, even in the knowledge that the new approaches will meet with resistance. The book's clarity and rich examples provide tools to be able to meet and accommodate resistance pragmatically and diplomatically. Mancuso appreciates that developers need to be able to find ways of being craftsmen that work for them in their particular situations. This flexibility and sensitivity gives the book the potential to improve every developer that reads it.