Software Development and Their Essences

By David Lambertson

1. Code and Fix

The Code and Fix model does address an essence. The one that led me to this decision is changeability.

While Code and Fix is not the best model, it is one that can adapt to changes quickly. Since the programmers are just typing code, running it and fixing it, adding new features are super easy.

1. Waterfall (Defined by Royce)

The Waterfall model does address an essence. The essence that led me to this decision is complexity.

The Waterfall method, while does not address changeability that well, it does handle complexity. Since you move through the model in different steps, requirements, design, code, test, and so on; the system could have a great degree of complexity. By taking each step one at a time you are able to slowly handle the complexity of the system.

1. Evolutionary Prototyping/Delivery

Through the changeability essence, I believe the Evolutionary Prototyping/Deliver model passes.

I believe this because even though you set things before you start, since you are either prototyping or delivering often, it’s easier than waterfall to add a new requirement or feature into the design. This leads this model to be changeable and flexible.

1. Spiral

The Spiral model does address an essence through the conformity essence.

I think this because with the Spiral model always comes back to review requirements and risks. Through it coming back and reviewing, the team using the model will be able to conform to the new requirements, new risks or even new challenges that have presented themselves. They can look at the big picture with a refreshed idea and make the necessary changes to the project.

1. Extreme Programming (XP)

The Extreme Programming model address an essence with the conformity essence.

The Extreme Programing model addresses the conformity because Extreme Programming involves pair programming. No programmer will work or should work alone while using this model. Pairs come together to work on the code, always having a fresh mind to look at the problem at hand in a different angle. Through this, the code is not a single creation, but the creation of multiple hands and thoughts.

1. Cleanroom Software Engineering

The Cleanroom Software Engineering model addresses the essence of complexity.

Through the Cleanroom Software Engineering model, programmers are creating code and software that has a certain level of reliability. Because of this certain level of reliability, it is easier to create complex software and code. Programmers are able to have complex algorithms and complex solutions to their problems and not have it hinder or not work with other parts of the project.

1. Automated Software Synthesis

The Automated Software Synthesis addresses an essence with the changeability essence.

Through Automated Software Synthesis, developing code becomes a simple task of just giving the requirements of the project and what is wanted and it will give back code. If there is a new requirement, you can just add it to the list and reproduce the code. Through this, it is quite easy to change any software.