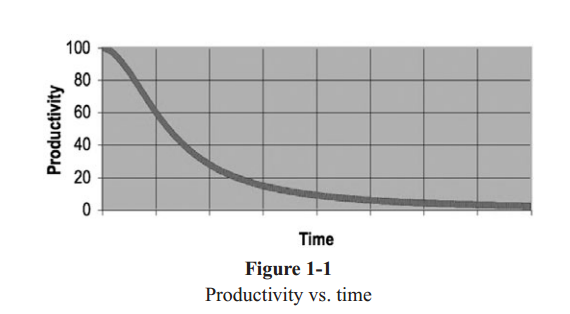
* El fin del código está cerca, pronto todo el código será generado en vez de escrito.
* Crees que aprender sobre el código ya no es necesario y en cambio nos deberíamos enfocar en los modelos y requerimientos.
* Los programadores no serán necesarios y generaremos los programas en base a los requerimientos.
* Alguna vez has tenido algún impedimento por culpa de mal código? Por culpa de tu propio código?  
  (PRODUCTIVIDAD)
* Conoces alguna empresa que haya caído o desaparecido por culpa del mal código.  
  (HISTORIA DE LA DESAPARICIÓN DE UNA EMPRESA)
* Con cual definición de Clean Code se identifica y crear su propia definición?

## Productividad

If you have been a programmer for more than two or three years, you have probably been significantly slowed down by someone else’s messy code. If you have been a programmer for longer than two or three years, you have probably been slowed down by messy code.

The degree of the slowdown can be significant. Over the span of a year or two, teams that were moving very fast at the beginning of a project can find themselves moving at a snail’s pace. Every change they make to the code breaks two or three other parts of the code. No change is trivial. Every addition or modification to the system requires that the tangles, twists, and knots be “understood” so that more tangles, twists, and knots can be added.

Over time the mess becomes so big and so deep and so tall, they can not clean it up. There is no way at all. As the mess builds, the productivity of the team continues to decrease, asymptotically approaching zero. As productivity decreases, management does the only thing they can; they add more staff to the project in hopes of increasing productivity. But that new staff is not versed in the design of the system. They don’t know the difference between a change that matches the design intent and a change that thwarts the design intent. Furthermore, they, and everyone else on the team, are under horrific pressure to increase productivity. So they all make more and more messes, driving the productivity ever further toward zero.

  
(Ciclo vicioso de menos productividad)

## XQ Clean Code?

Let’s say you believe that messy code is a significant impediment. Let’s say that you accept

that the only way to go fast is to keep your code clean. Then you must ask yourself: “How do I write clean code?” It’s no good trying to write clean code if you don’t know what it means for code to be clean!

The bad news is that writing clean code is a lot like painting a picture. Most of us know when a picture is painted well or badly. But being able to recognize good art from bad does not mean that we know how to paint. So too being able to recognize clean code from dirty code does not mean that we know how to write clean code!

When hand-washing was first recommended to physicians by Ignaz Semmelweis in 1847, it was rejected on the basis that doctors were too busy and wouldn’t have time to wash their hands between patient visits.

Writing clean code requires the disciplined use of a myriad little techniques applied through a painstakingly acquired sense of “cleanliness.” This “code-sense” is the key.

Some of us are born with it. Some of us have to fight to acquire it. Not only does it let us see whether code is good or bad, but it also shows us the strategy for applying our disci-pline to transform bad code into clean code.

A programmer without “code-sense” can look at a messy module and recognize the mess but will have no idea what to do about it. A programmer with“code-sense” will look at a messy module and see options and variations. The “code-sense” will help that pro-grammer choose the best variation and guide him or her to plot a sequence of behavior preserving transformations to get from here to there.

In short, a programmer who writes clean code is an artist who can take a blank screen through a series of transformations until it is an elegantly coded system.