Leandro Maglianella (K1833503)
Kingston University

Essay on Wicked Problems

A wicked problem is defined as a problem that can not be definitively solved. In general, the development of any software respects all the characteristics to be considered a wicked problem, for example every time a code is written it will never be universally correct because for every problem there are infinite ways to solve it and each of these is imperfect.

When the development of a software is started, it is not possible to immediately establish in which difficulties we will encounter, complications can come from any area and for any reason, often even from codes that are believed to be infallible and that have worked in the past. In fact, working software is not reliable, it never implies that it will continue to function indefinitely and even the most experienced programmer in the world can never be completely certain of its own software.

No code is infallible because no human being is, even a seemingly perfect software is never globally usable due to the enormous amount of different types of computers, processors and operating systems; even assuming the existence of an entirely bug-free software, it will soon become obsolete as there will always be some feature to be added. For this reason, software development is not an activity to be followed iteratively step by step, but it is a cyclical work of design and development.

There are many past examples of software being developed by large teams but they had to be abandoned because they were impossible or too expensive to build, for instance one of the major projects abandoned in the history of the United Kingdom is a system for electronic care records called NHS Connecting for Health whose cost is estimated to be around 20 billion pounds.

A further problem of software development that makes it a wicked problem consists of the multiple interdependencies and pre-existing platforms to which the code we write must support: a web application for example is constantly dependent on all the other components of the network with which it interacts and from across the internet in general.

In conclusion, it is the developer's task to understand that the challenges he faces every day are difficult and that he will often have to retrace his steps to correct and improve his software, without however frustrating himself in not finding a final and definitive solution.

References

Wicked problem (2019) Available at: https://en.wikipedia.org/wiki/Wicked_problem (Accessed: 14 October 2019).

Hartmann, B. (2013) Some Wicked Problems of Software Design. Available at: https://www.slideshare.net/bjo3rn/2013-1026wickedproblemswithnotes (Accessed: 14 October 2019).

Solving the Wicked Problem for Software Engineering (2015) Available at: http://blog.nawaratne.com/2015/08/solving-wicked-problem-for-software.html (Accessed: 14 October 2019).

Development is Inherently Wicked (2004) Available at: https://blog.codinghorror.com/development-is-inherently-wicked/ (Accessed: 14 October 2019).

Poppendieck, M. (2002) Wicked Problems. Available at: http://www.leanessays.com/2002/01/wicked-problems.html (Accessed: 14 October 2019).