Student : Tom BOURRET

Software Engineering : Biography of Larry Constantine

- *A short biography of the individual*

Larry LeRoy Constantine, born in 1943, is an American software engineer, professor in the Center for Exact Sciences and Engineering at the University of Madeira Portugal, and considered one of the pioneers of computing.

Graduated from Anoka High School in 1961, started his working career as a Technical Aid/Programmer at M.I.T. Laboratory for Nuclear Science in 1963. He is known for his contributions for numerous concepts and techniques forming the foundations of modern practice in software engineering and applications design and development.

He collaborated on several publications, like "*Sotfware to Use*", "*Modular Programming*" and "*Structured Design*". In this particular work, Constantine develop the concepts of structured analysis and its key features such as structure chart and data flow diagram. He started to work on structure design as undergraduate at MIT, and develop new concepts as parts of structure design in the next years, like cohesion and coupling, and developed methodologies like usage-centered design.

Constantine also developed methodologies that combine human-computer-interaction design with software engineering, like usage-centered design which is the topic of the book "*Software For Use*" he wrote with Lucy Lockwood in 1999. It has been an important contribution, even to human-computer interaction, well used in professional practice and the subject of academic study.

- *Discussing the work and impact of the individual*

In my opinion, the work of Larry Constantine on structured design and modular programming is an important basis of every important development projects today. Many of those concepts are still learned in computer science courses and are the basis of efficient and reusable code.

Modular programming, which is “software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules” (cf wikipedia page), is an efficient way to parallelize work on a project and produce reusable code. The negative part of modular programming is the upstream work to split a project in independant modules, but the result is objectively significant, it is much more efficient to maintain, to improve and to test.

In the same logic, structured analysis (or structured design) in general became extremely relevant with the prolifération of applications (mobile, tablets, computers, tv,...). Many people currently work to produce or improve those applications, and many of them do not produce or maintain code. In this situation, the communication is the only way to achieve to a result that is suitable for both developers and non-developers. And that’s how tools like structured design fits perfectly, making possible to transpose business requirements and specifications into computer programs.

The use of diagrams to describe structures, flow, contexts and procedures often can seem to be a long, expensive and non productive, but it consists in an important point : transform the complex reality of some activities, especially for developers who may not know anything about a specific domain they are developing for, into a visible, understandable diagram. And it has many advantages : it could reduce the risks of misunderstanding some parts, be easier to transpose into a program, and also be easier to present the project to a newcomer.

Thus, the human-computer work of Larry Constantine, with notably usage-centered design which focus on the user intentions and roles when designing an interface, brings real efficient tools for some types of websites or application in my opinion. When developers or designers work new types of application, they might not think of reals expectations of final users. Usage-centered design consist of asking directly to users (using an old version or just asking futur users) at each steps. The idea is to adapt the application to the users instead of impose an interface to the user.

In my opinion, this approach is extremely relevant in the current situation, with a lot of competition and similar applications, because it incite developers and designers to fit with users wishes to attract and keep as much customers as possible.

Finally, I'd say that Larry Constantine's works are as relevant today as ever, and brought some of the most important way of thinking an application, and present a huge basis for any developer with some ambition.