Architectural Pattern

Thomas Traxler

May 23, 2014

Contents

1 Architecture

"The word conjures up images of flying buttressed cathedrals and Doric columns, even among computer cognoscenti. Yet more than years have passed since computer science appropiated the term for a new meaning. Back then most people thought 'software developement' meant 'writing code', but some in the software developement community knew better. They observed people attempting ever-larger software systems, and they saw how traditional ad hoc development approaches were failing them with alarming regularity.
'Large' systems are inherently larger than one person can build. So to build them, developers had to foste discipline and cooperation. They had to scope out their system ahaed of time and break it up into manageable pieces. They needed ways of specifying what the pieces did and how they communicated with other pieces. They had to define abstractions that gave programmer and maintainersinsight into the system's workings. They had to be creative in the large, not just clever in the small. In short, they needed a notion of overall design apart from implementation - *software architecture* "[LS96]

I wanted to start with this cite of the beginning of the Chapter 'Architectural patterns' from the book pattern languages of programm design 2 because it describes the problematic in nowerdays software developement in combination with the term *Architecture* in it's classic meaning as good as nobody else. It's written nearly twenty years ago (1996) and lost nothing of it's rightfullness. Programms are still getting ever bigger as then (See Figure 1.1) and need an cleare structure, an architecture as much as than and software architectures are often designed for one specific programm, still there are patterns in this architectures. Patterns that can be reused, patterns which proofed themselfs in many other programms, patterns which can be of great help for the architects and patterns which will be described in the following chapters.
[BMR+96][Fow02]

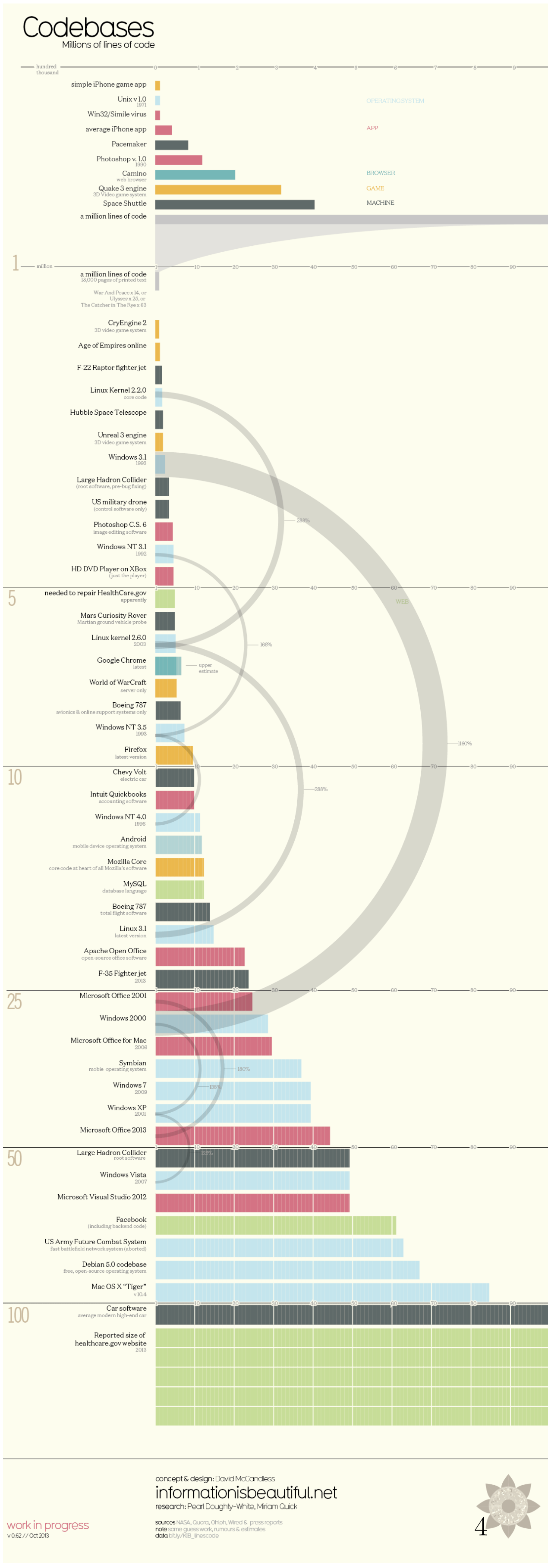

Figure 1.1: A million lines of code in various programms

2 Pattern

2.1 Pipeline (and Filters)

2.2 Data Abstraction

2.3 Communication Processes

2.4 Implicit Invocation

2.5 Repository/Blackboard

2.6 Interpreter

2.7 Main Program and Subroutines

2.8 Layered Architecture

2.9 Broker?

2.10 Microkernel

2.11 Reflection?

2.12 MVC?

2.13 Presentation-Abstraction-Control

2.14 Evolution, Architecture, and Metamorphosis?

2.15 Patterns for Mapping to Relational databases?

2.15.1 Gateway

2.15.2 Data Mapper

Bibliography

[BMR^{+}96] Buschmann, Frank ; Meunier, Regine ; Rohnert, Hans ; Sommerlad, Peter ; Stal, Michael: *Pattern-Oriented Software Architecture - A System of Patterns*. 1996. – 476 S. – ISBN 0471958697

[Fow02] Fowler, Martin: *Patterns of Enterprise Application Architecture*. Bd. 48. 2002. – 560 S. `http://dx.doi.org/10.1119/1.1969597`. `http://dx.doi.org/10.1119/1.1969597`. – ISBN 0321127420

[LS96] Lavender, R G. ; Schmidt, Douglas C.: Pattern Languages of Program Design 2. In: *Pattern Languages of Program Design 2*. 1996. – ISBN 0–201–895277, S. 483–499