

# FUTURE-PROOF SOFTWARE

Lecture Wintersemester 2016/17

Prof. Dr. Frank J. Furrer



[© rf123.com with permission]

## Context

Today we live in a **software-world**. Most of our products and services are based on software. Our dependence on software is nearly total.

This lecture focusses on a specific type of software: Mission-critical, business-essential and long-lived software. **Mission-critical** means that a malfunction or unavailability of the software inhibits the use of the product or service and may cause damage, accidents or loss. **Business-essential** implies that the software is key to the success and development of a company or organization. Finally, **long-lived** signifies that the software-system must be maintained, extended and evolved over many years, possibly decades.

Mission-critical, business-essential and long-lived software is essential for a very broad range of applications, such as cars, aeroplanes, ground transportation systems, the energy distribution, the telecommunications infrastructure, financial services, traffic control, and many more. Our world cannot function properly without this type of **future-proof software**.

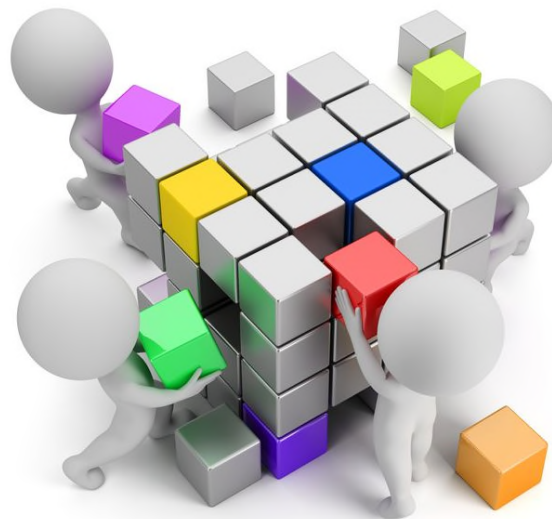
## Software Architecture

Future-proofness of software is the result of its **architecture**. Only a well defined, adequate and consistently evolving architecture assures the necessary functional and non-functional properties. Fortunately, systems architecture has matured from a “black art” to a well founded system engineering discipline today. Architecture is taught and implemented using **architecture principles**. This lecture focuses on the important architecture principles for future-proof software – presenting the “IT architects toolbox”, focused on **agility** and **resilience** of software systems.

## Software Engineering

Software – and the ability to produce software – has become a key competitive factor in most industries. The fundamental quality properties: Business value, agility and resilience of software often decide over success or failure of a product or service. In order to maintain a **competitive industry**, we need future-proof software.

Future-proof software is specified, architected, designed, produced and evolved by **software engineers**. For future-proof software specialized skills – both hard skills and soft skills – are required. Those will also be taught in this lecture, thus making the participants valuable members of their future software teams in industry.



[© rf123.com with

permission

## Lecture Dates

Time 3.+4. DS (11:10 – 12:40 and 13:00 – 14:30) in room APB/E010, on the following dates:

Date	Topic
Wed, 12. Oct. 2016	Introduction
Wed, 26. Oct. 2016	Managed Evolution of Software
Wed, 9. Nov. 2016	Architecting for Agility (1)
Wed, 23. Nov. 2016	Architecting for Agility (2)
Wed, 7. Dez. 2016	Architecting for Agility (3)
Wed, 21. Dez. 2016	Architecting for Resilience (1)
Wed, 18. Jan. 2017	Architecting for Resilience (2)
Wed, 1. Feb. 2017	Skills and Personality of the Future-Proof Software-Engineer

## Exams and ECTS

Attendance to the lecture results in 3 ECTS credits. An attendance certificate (Sitzschein) or a graded exam (10 minutes, oral) is possible.

## Contact

For more information please contact Prof. Dr. Frank J. Furrer at [frank.j.furrer@bluewin.ch](mailto:frank.j.furrer@bluewin.ch) or go to the lecture-website. All lecture slides will be made available on the lecture-website before the corresponding lecture data.