INSTITUT FÜR SOFTWARE

HSR HOCHSCHULE FÜR TECHNIK RAPPERSWIL

INFORMATIK

From Snake-Oil and Silver-Bullets to

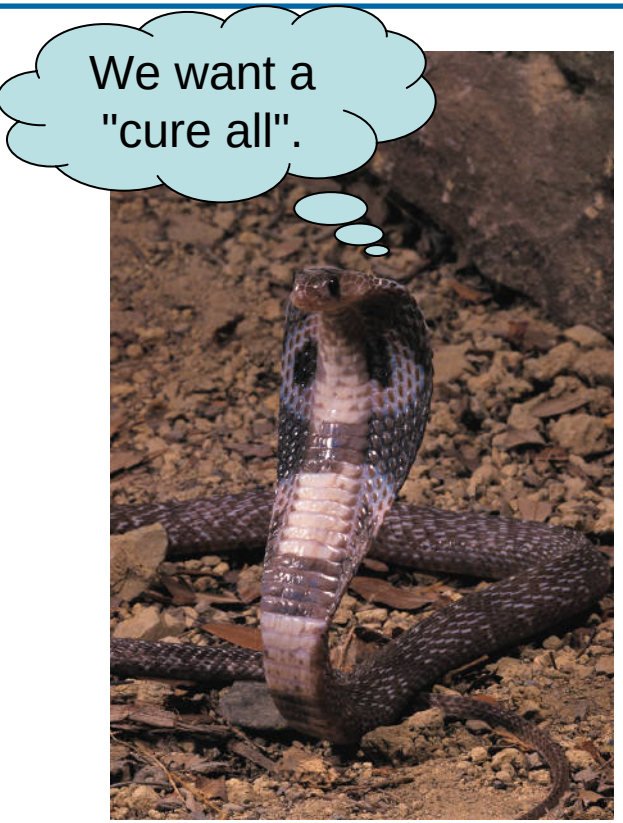
*Agile-Incremental-Iterative-Pattern-oriented**

*** sing to the tune of "Mary Poppins"**

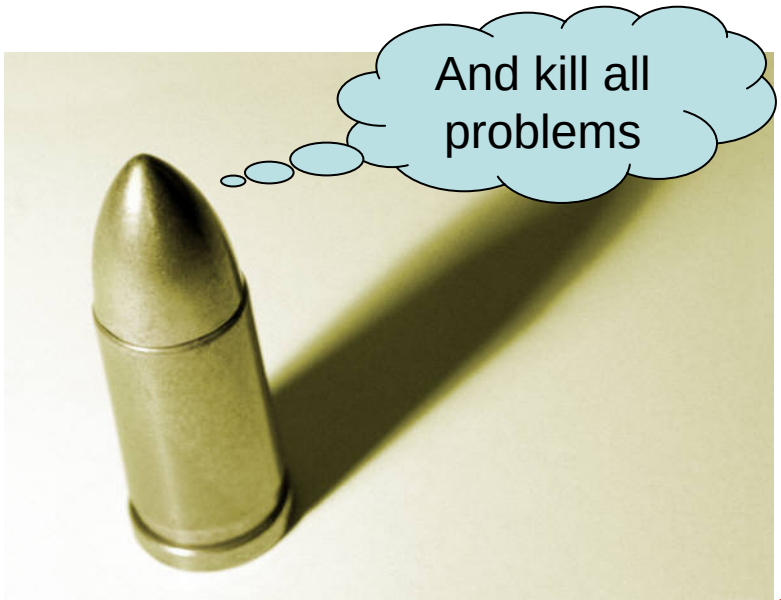
Peter Sommerlad, Programmer, Prof, IFS, HSR

Snake-Oil and Silver Bullets

INSTITUT FÜR SOFTWARE



- **Acronym Jungle**
  - CASE, OOP, CMM, SGML
    - more modern:
  - XML, EJB, .NET, UML, MDA
- **Technology Overload**
  - C++
  - Corba
  - Java
  - C#
  - VB



COMPLEXITY

Complexity is one of the biggest problems with software.

It is much easier to create a complicated "solution" than to really solve a problem.

Reasons for Complexity

INSTITUT FÜR SOFTWARE

- **Young Guns**
  - "Hey, I learned so many complicated things during my time at HSR, I want to use it now!"
  - Coolness is important!
  - Complex stuff is cool!
  - Over-Engineering
- **Amateur Programmers**
  - "I don't know how it works, but I made it run."
  - Programming by Coincidence
  - No idea of Abstraction
  - Copy-Paste Reuse
  - Under-Engineering
- **Media**
  - "There is this brand new stuff called XYZ, we tell you how to achieve productivity increasement with it"
  - sells only "newest" stuff
- **Consultants**
  - "We must use XYZ for your problem" ... thinking "because it gives us more billable hours"
- **Resume-oriented Developer**
  - "I'll use this cool new stuff, because it looks good on my resume"

SIMPLICITY

We need to value Simplicity much higher.

Our software needs to be simpler to solve more complex problems.

How and why?

INSTITUT FÜR SOFTWARE

- **Reduce your code size to 10%**[1]
  - Manageability
  - Extendability
  - Maintainability
  - Quality
  - Testability
  - …ility

1. NOT: "by 10%"!

Curing unnecessary Complexity?

INSTITUT FÜR SOFTWARE

- **How to simplify Software?**
  - Abstraction
  - Refactoring
  - Generators
  - Data-Driven Development
- **What do I need? (Excerpt)**
  - Unit-Testing
  - Build- and Test-Automation
  - Iterative-Incremental Development
    - with (very) short iterations
  - Courage to Simplify, Reflect and Refactor

Do we need it?

tick for yes (blatantly adapted from "The Joel Test")

INSTITUT FÜR SOFTWARE

1. We use **Version Control**.
2. We have a **1-Step Build Process**.
3. We have a dedicated **Continuous-Build** Machine running.
4. We have a **Collaboration** tool/Wiki.
5. We **fix Bugs before** we implement new features.
6. We write **Automated Tests** before we code or fix bugs.
7. We have defined **short Iterations** and our schedule is up-to-date
8. We Design and **Specify succinctly** what we want to test and develop
9. We **Program in Pairs**
10. We have **Testers** in addition to full Test Automation.
11. We hire only **candidates** that demonstrate their **programming** ability.
12. We have (hallway) **Usability Testing**.