

Software Engineering

Romain Lenglet

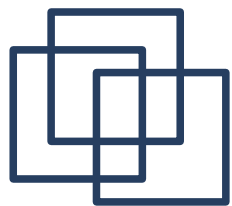
ロマン ラングレ

JSPS Post-doctoral Fellow

Pr. Chiba Shigeru Group
Tokyo Institute of Technology

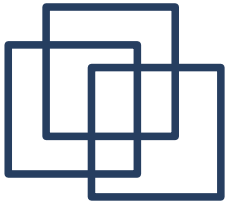
2005-10-08

Takasaki High School
International Science Camp



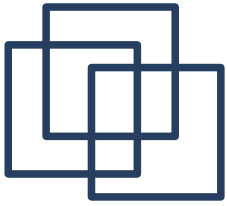
Quick introduction

- I am French, 27 years old
- I come from Grenoble (South-East of France)
- In Japan since December 2004
- Post-doctoral researcher in the Tokyo Institute of Technology

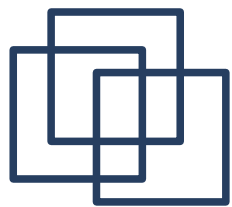


Plan

- My studies in France
- My research in software engineering
- Everything will be in English
- Please interrupt me to ask questions!
- Please ask Chiba! ^_^



My studies in France



Universities in France

“Old” French system

New European system

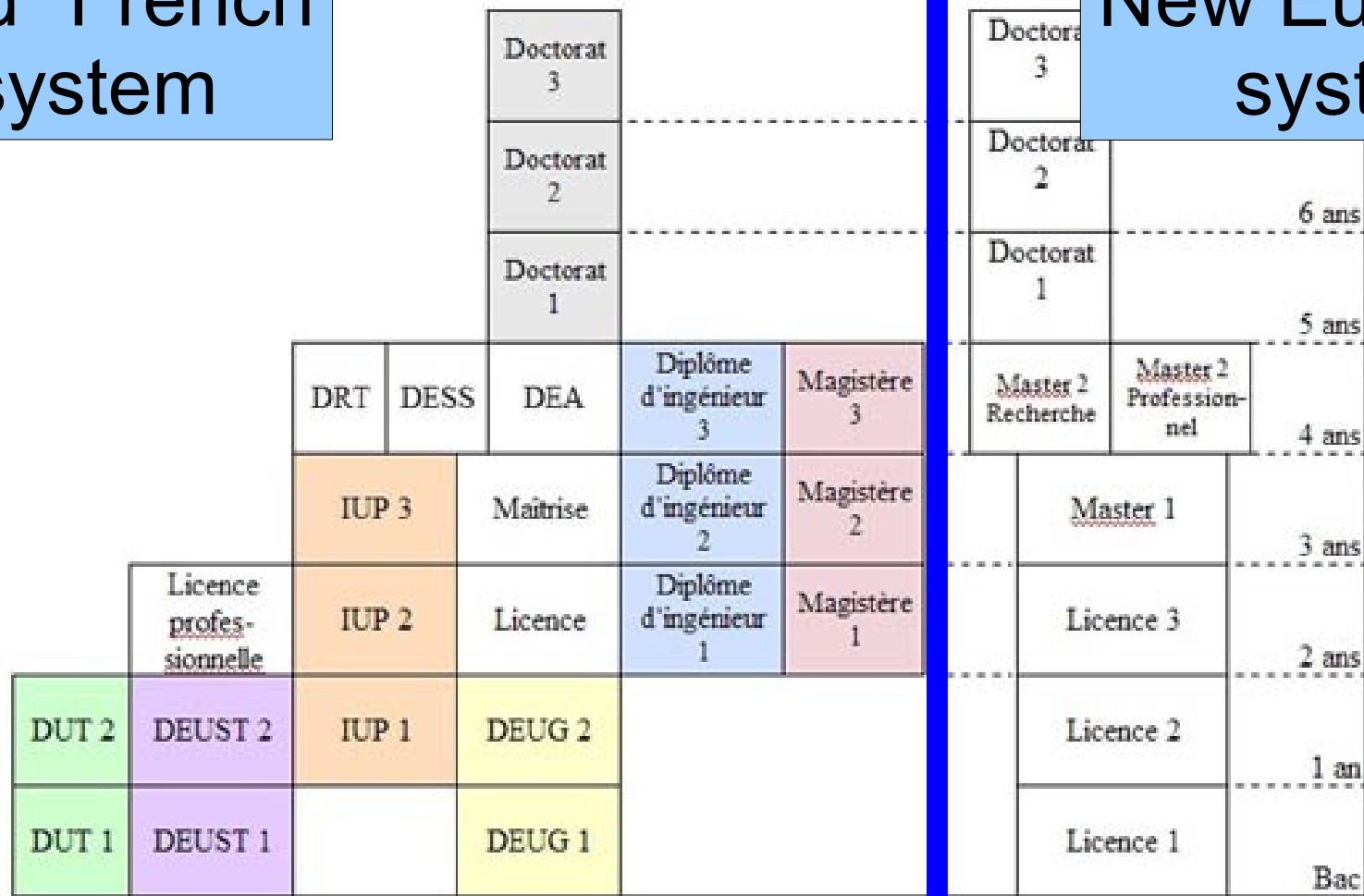
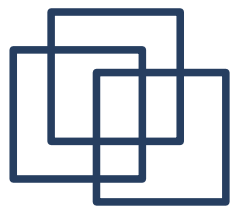
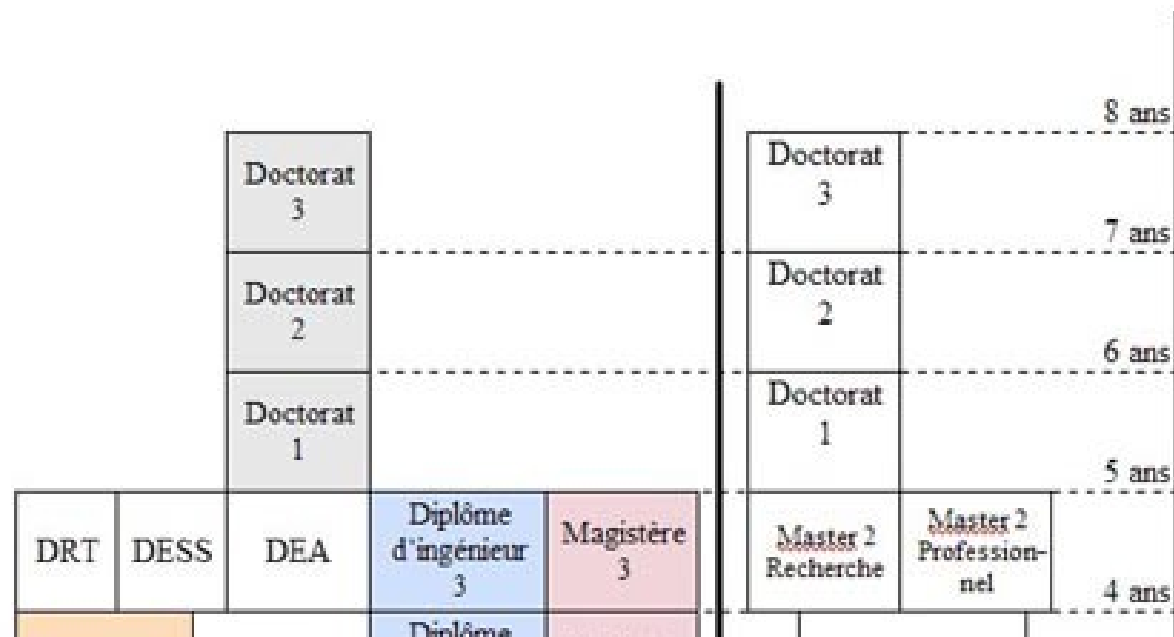


schéma actuel d'organisation des études

nouveau schéma



Universities in France

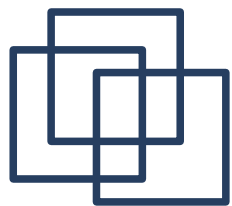


Baccalauréat:

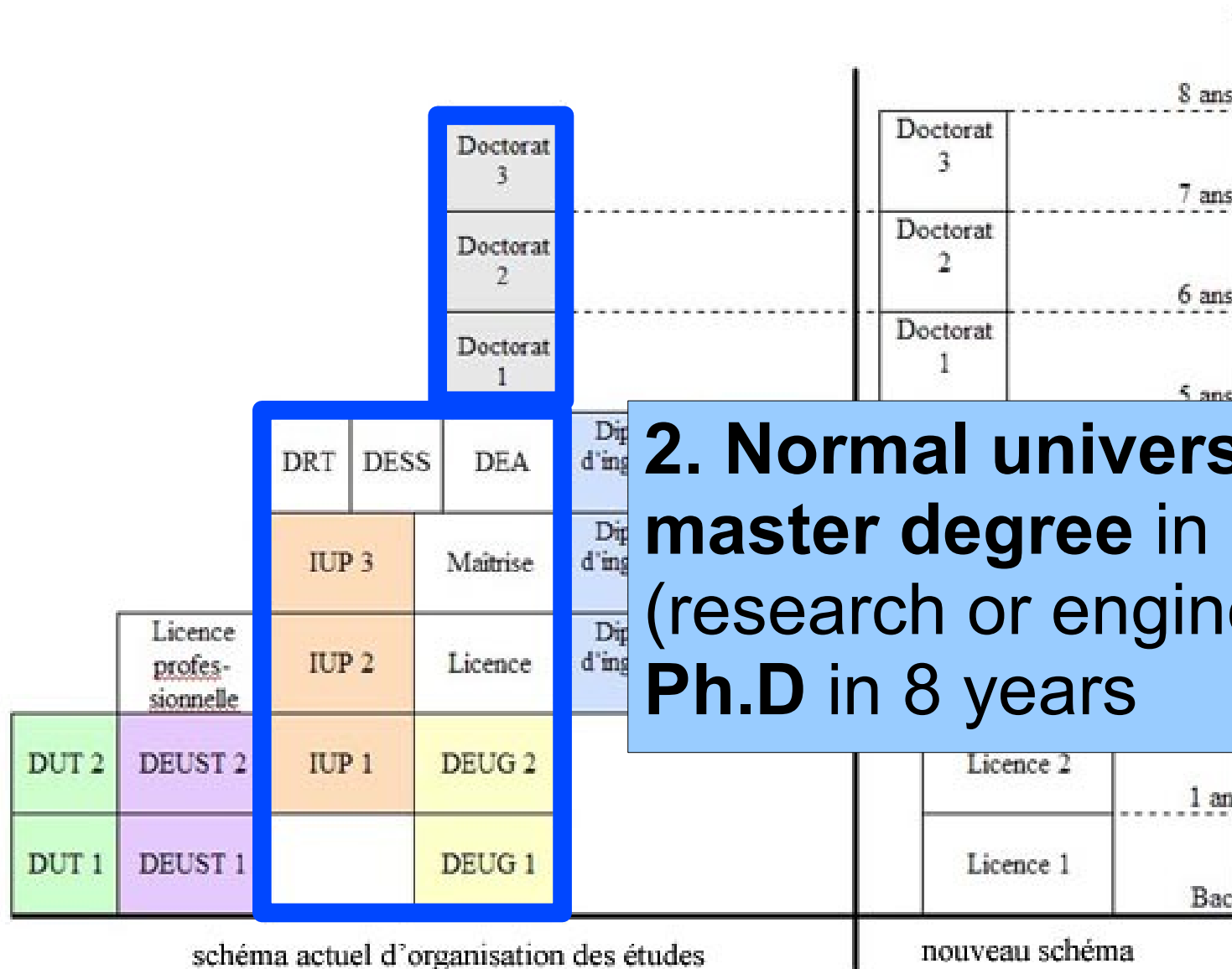
Common national exam before entering university, in everything: history, geography, mathematics, physics, biology, French, English, etc.

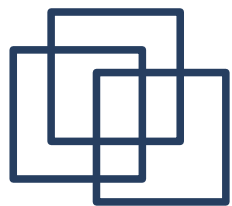
schéma actuel d'organisation des études

nouveau schéma



Universities in France

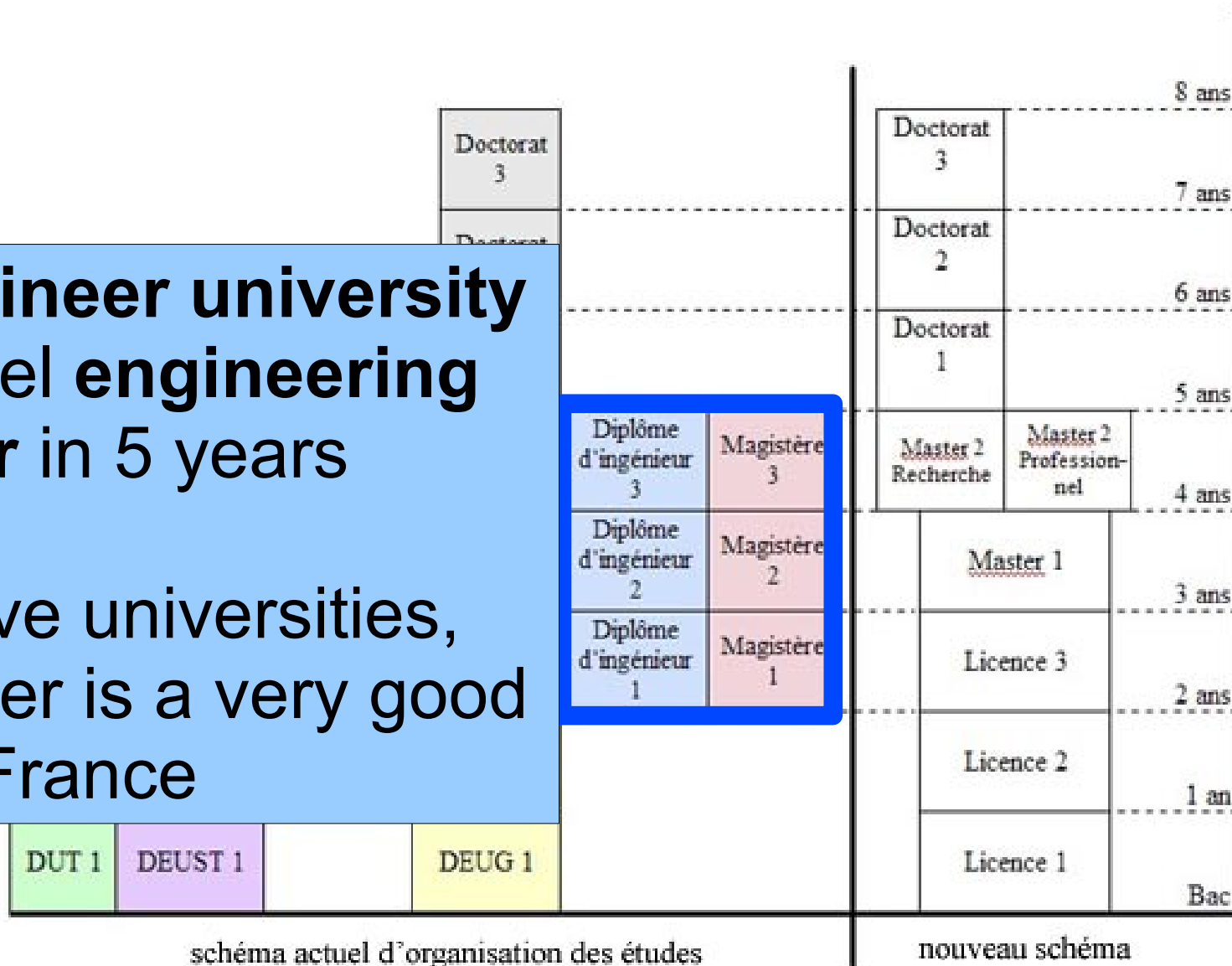


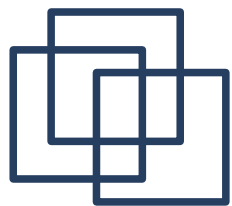


Universities in France

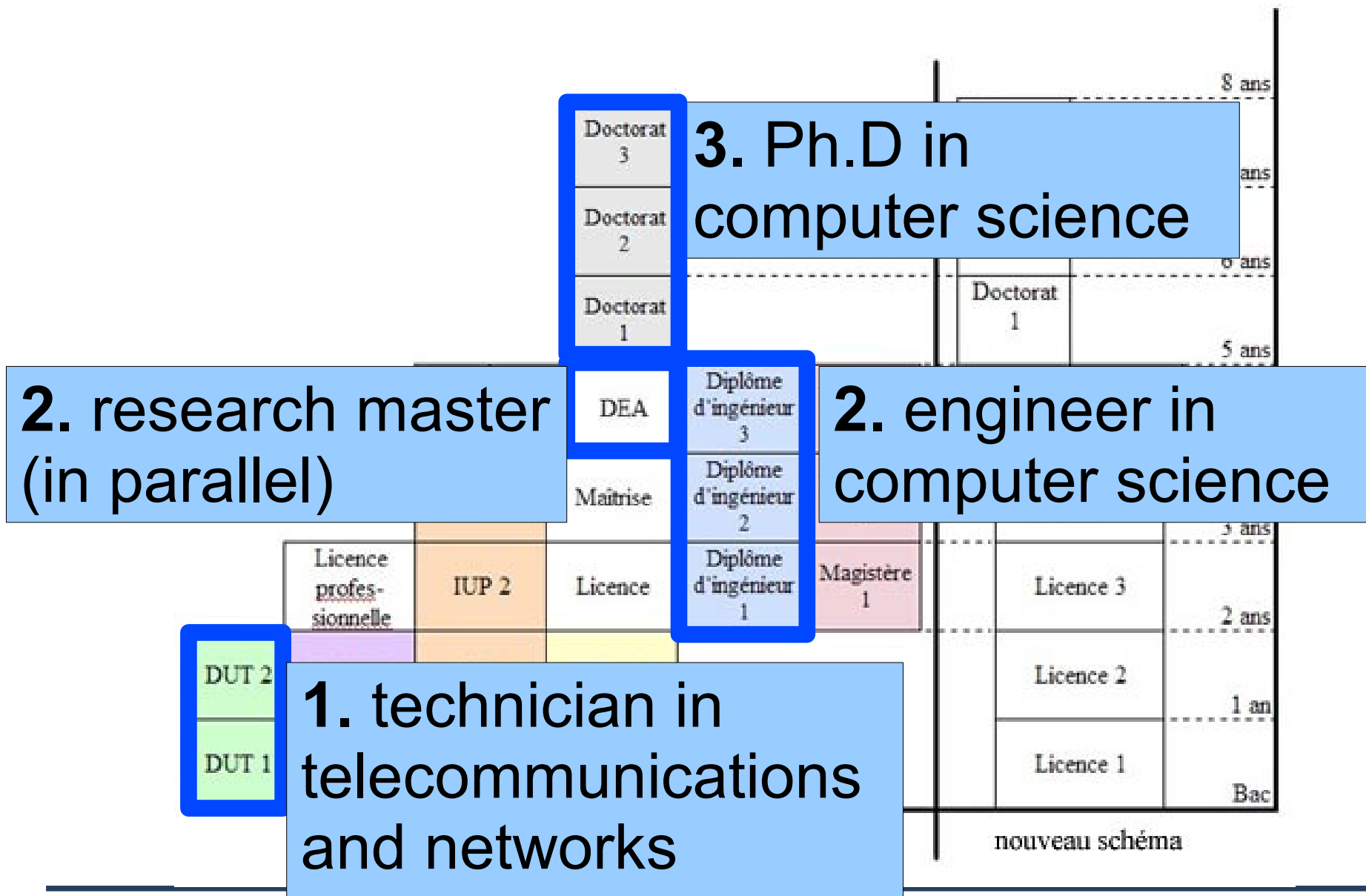
**3. Engineer university
top-level engineering
master in 5 years**

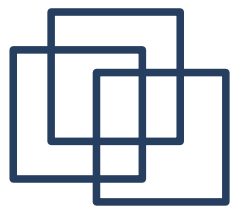
selective universities,
engineer is a very good
job in France





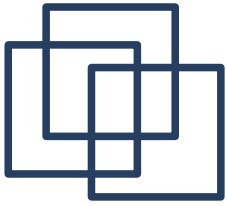
My studies in France





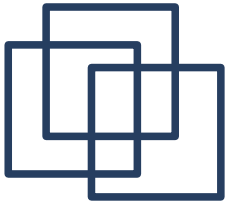
My Ph.D

- 3 years: 2001-2004
- Usually, Ph.D students work in a public laboratory
- My case was special:
 - Ph.D in **private company: France Telecom**
 - **Employed as a research engineer** for 3 years
 - Contract between the government, a private company, a public laboratory, a university, and a Ph.d student (like me!)



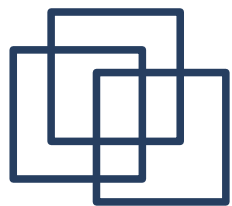
My Ph.D

- My two ex-laboratories, in Grenoble
 - France Telecom R&D
 - France Telecom is France's biggest telephone operator: 250 000 employees
 - Big research centers: 4200 researchers and engineers (biggest research center in Europe for computer science and telecoms)
 - INRIA (インリヤ)
 - France's public national institute for research in computer science and automatics
 - 3500 employees (2700 researchers)



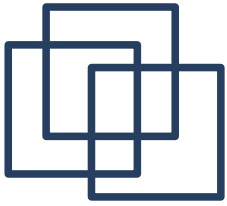
My Ph.D

- Very interesting environment
 - Many researchers to talk with
 - (More researchers than students)
 - Work in groups / projects
 - Many different research domains
 - Good salary ^ _ ^
- Very different from Japan
 - One lab = 1 professor + many students
 - Laboratories are isolated

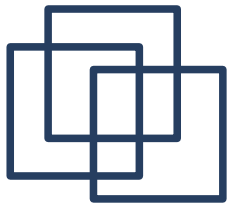


My post-doc in Japan

- Since December 2004, for 2 years
- What is a post-doc?
 - Research just after Ph.D
 - **Good for employment** afterwards: shows that I can adapt, I can lead a research project
- Why Japan?
 - Attracted by Asia
 - Good work conditions (money for research)
 - Good opportunity (contacts in Japan...)

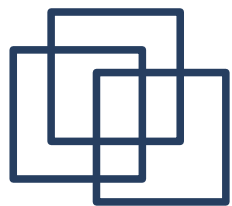


My research



Software Engineering

- There are two kinds of sciences:
 - Natural Sciences
 - Goal: to **explain nature**, to discover natural laws
 - **Sciences of Design**
 - Goal: to solve problems by **creating new artifacts** (= artificial things)
- Software Engineering is a Design Science
 - Software programs = artifacts
 - **How to create, manage & execute software programs**



Software Engineering

Requirements: “what we want / need”



“I want an email program with BIG buttons (2cm high)”

Software design

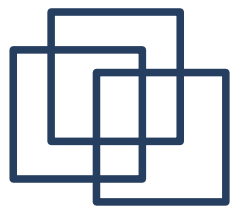


Good design: the program has big buttons

Software system



SEND



Software Engineering

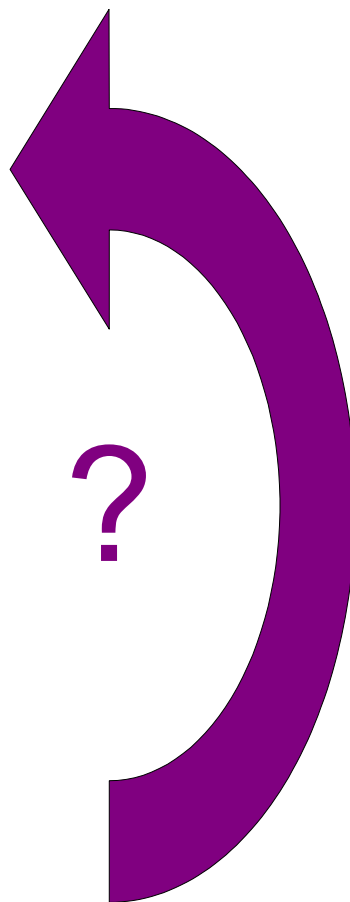
Requirements: “what we want / need”



Software
design

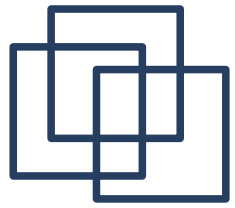


Software
system



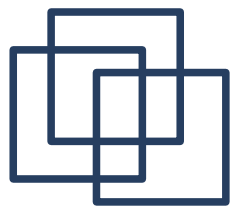
Problem: how to make
sure that the software
does what we want?

This is the problem
addressed by
Software Engineering



Software Engineering

- Purpose of Software Engineering:
 - How to have **good software design**
- Many viewpoints of that problem:
 - Theoretical: very mathematical
 - Technical: operating systems, compilation...
 - High-level: **Software Architecture**
 - Subdomain of Software Engineering

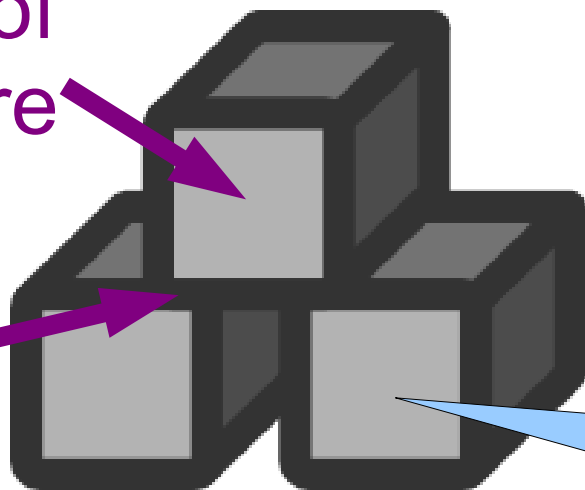


Software Architecture

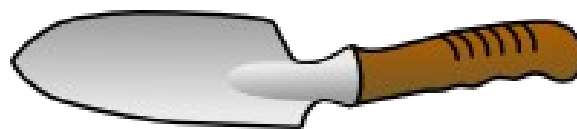
- Hide details of programs (the code...)
- Software \approx buildings

bricks of software

glue



construction tools

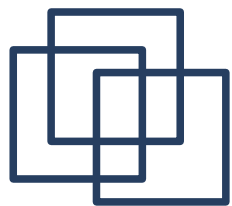


For example:
brick = Java program

construction plans

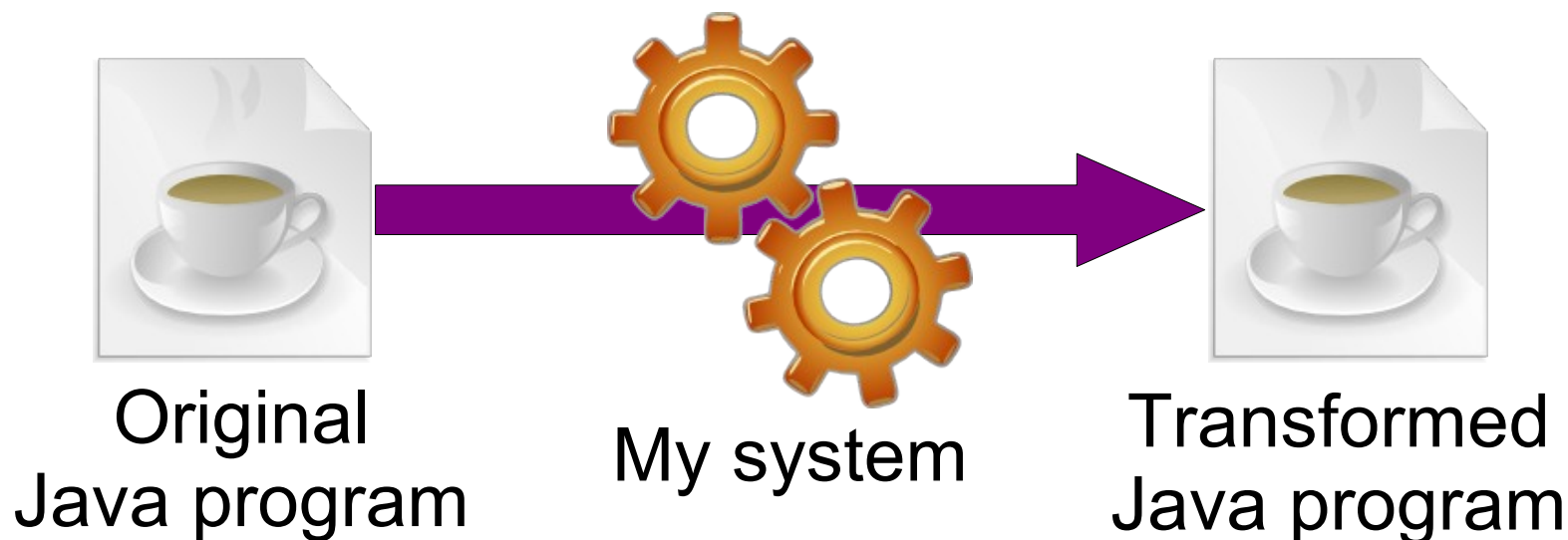


- What are the bricks / tools / glue?
 - How to define “construction plans”?
-

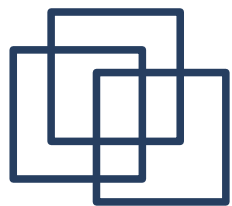


My Ph.D, in France

- Short summary of my Ph.D
 - I have designed a program that **automatically transforms Java programs**



- This is a kind of “construction tool” 



Systems Management

Requirements: “what we want / need”



What happens if what we want, or the environment **change during execution?**

Software design

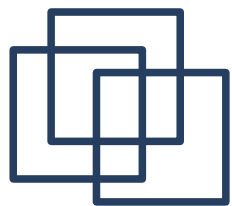
“Finally... I want **SMALL** buttons in the program (0.5cm high)”

Software system



The software does no more do what we want!

SEND



Systems Management

Requirements: “what we want / need”



We need to add another system, to adapt the system to changes

Software design

The software does what we want, now!

SEND

Management system

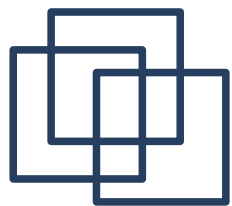
Software system



Management



Make buttons smaller



Systems Management

Requirements: “what we want / need”



How are we sure that
management has good effects?
= How are we sure that the system
does what we want, again?

???

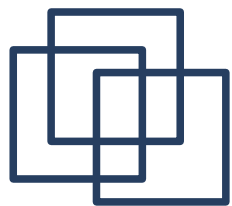
This is an original problem!

Software
system



Management





Systems Management

Requirements: “what we want / need”



Bad effect = bad management!

SEND

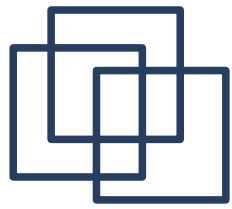
Software system



Management



Make buttons bigger



Systems Management

Requirements: “what we want / need”



My research:
find a **method to design**
management systems
that have good effects

OK

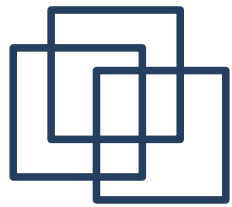
**Management
system**

Software
system



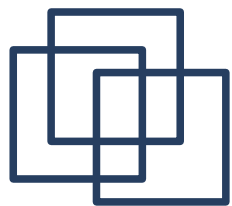
Management





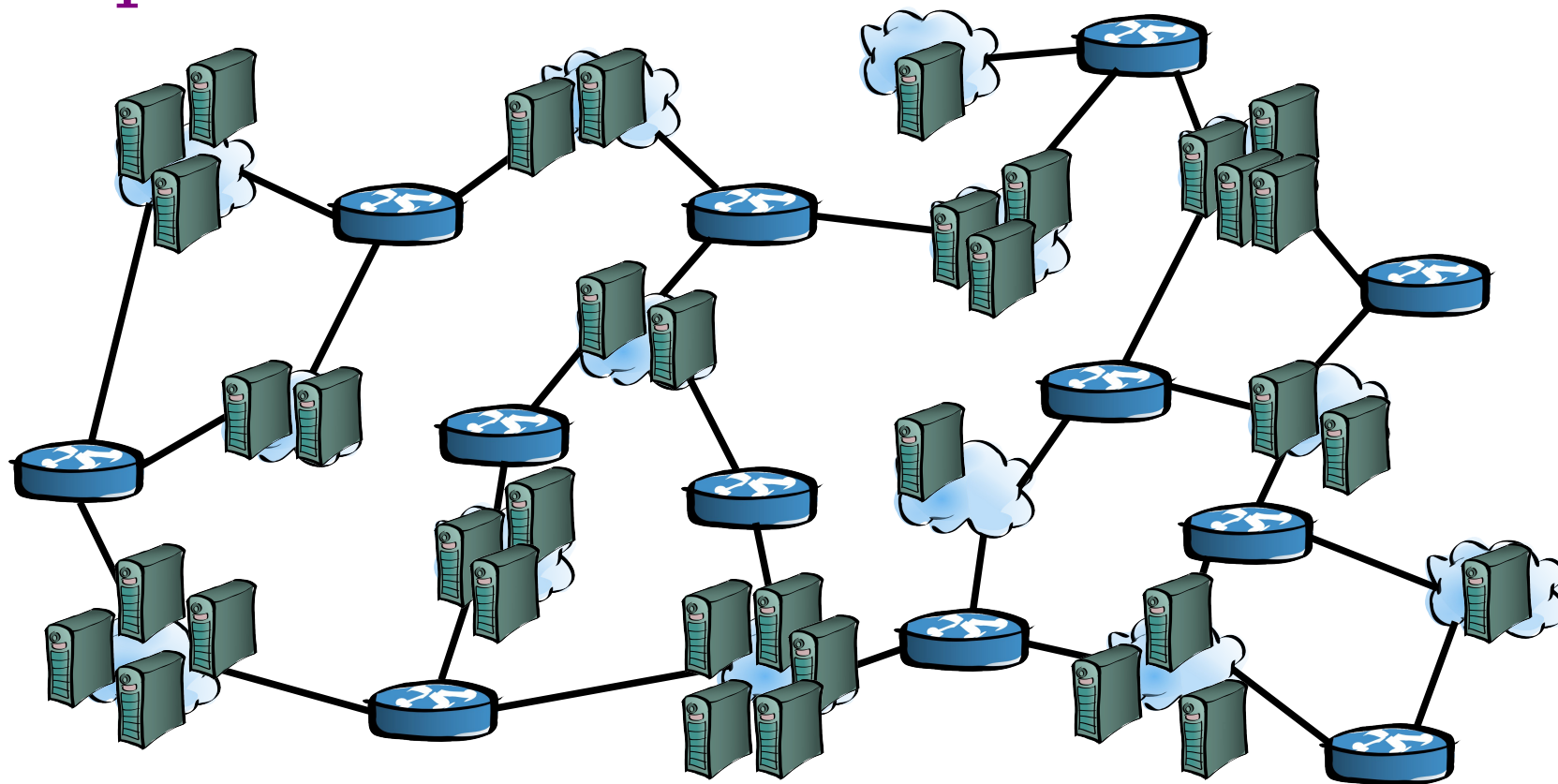
Systems Management

- Management is a decision problem
 - What action is the best?
 - Make buttons smaller
 - Make buttons bigger
 - How to choose the best action?
- My solution
 - Apply existing decision methods that work very well already in the “real world”
 - To manage companies, hospitals, cities...
 - Multiple-Criteria Decision Methods (MCDM)

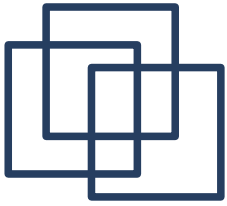


Systems Management

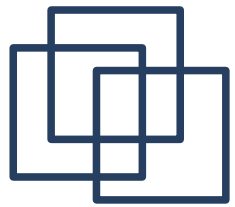
- My solution can manage **very big, very complex software**



- For example: telecom network
-

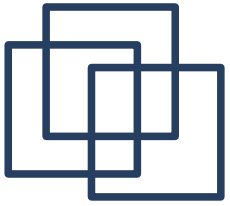


Concluding remarks



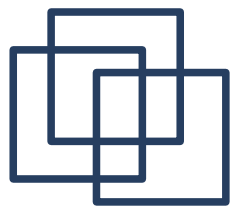
What I like in my research

- Science of Design = **creativity**
 - We create software: creativity is essential in our domain
 - It is very easy to create software, easy to be creative
- Science with **applications**
 - To prove that our ideas are good, we
 - **Develop software** that apply our ideas
 - **Share software** as Open Source / Free Software
 - Our software is really used worldwide



Trends

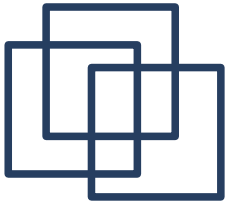
- Change in focus
 - Think of SE really as a Science of Design
 - From: **what** is software (languages, tools...)
 - To: **how** to make sure that software is useful
- **Individual people are the most important**
 - Software must solve problems of humans
 - Developers / designers are also important: must be creative, be able to communicate...
 - This is a difficult cultural change



My future research

- Apply my research in another domain
 - Computer-aided surgery
 - Software useful in reality
 - Human-focused
 - Patients
 - Multi-disciplinary domain: interesting relationships between researchers
 - They really need specialists in Software Architecture





If you are interested...
Please come visit us!

Tokyo Institute of Technology
Open Campus
22-23 October 2005