SECR
Software Engineering Conference Russia
St. Petersburg October 20 – 22 2017

LIBERAL ARTS IN A DIGITALLY TRANSFORMED WORLD: REVISITING A CASE OF SOFTWARE DEVELOPMENT EDUCATION

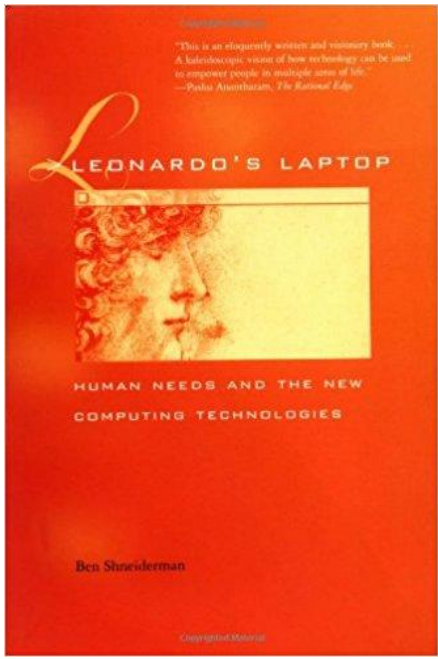
Evgeny Pyshkin

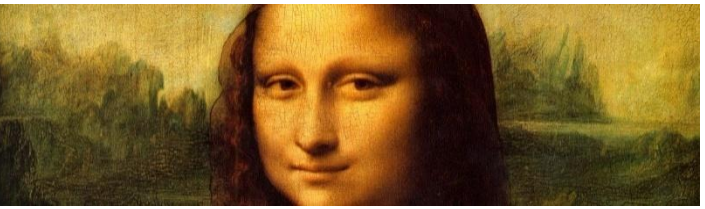
University of Aizu

会津大学

What is Human-Centric Computing

- Ben Shneiderman's *"Leonardo's Laptop: Human Needs and the New Computing Technologies"**
  - The focus is shifting from what computers can do to what users can do
  - A key transformation is to what he calls "universal usability," enabling participation by **young and old, novice and expert**, able and disabled
- Human Centered Computing (HCC) aims at bridging the gaps between the disciplines involved with the design and implementation of computing systems that support people's activities



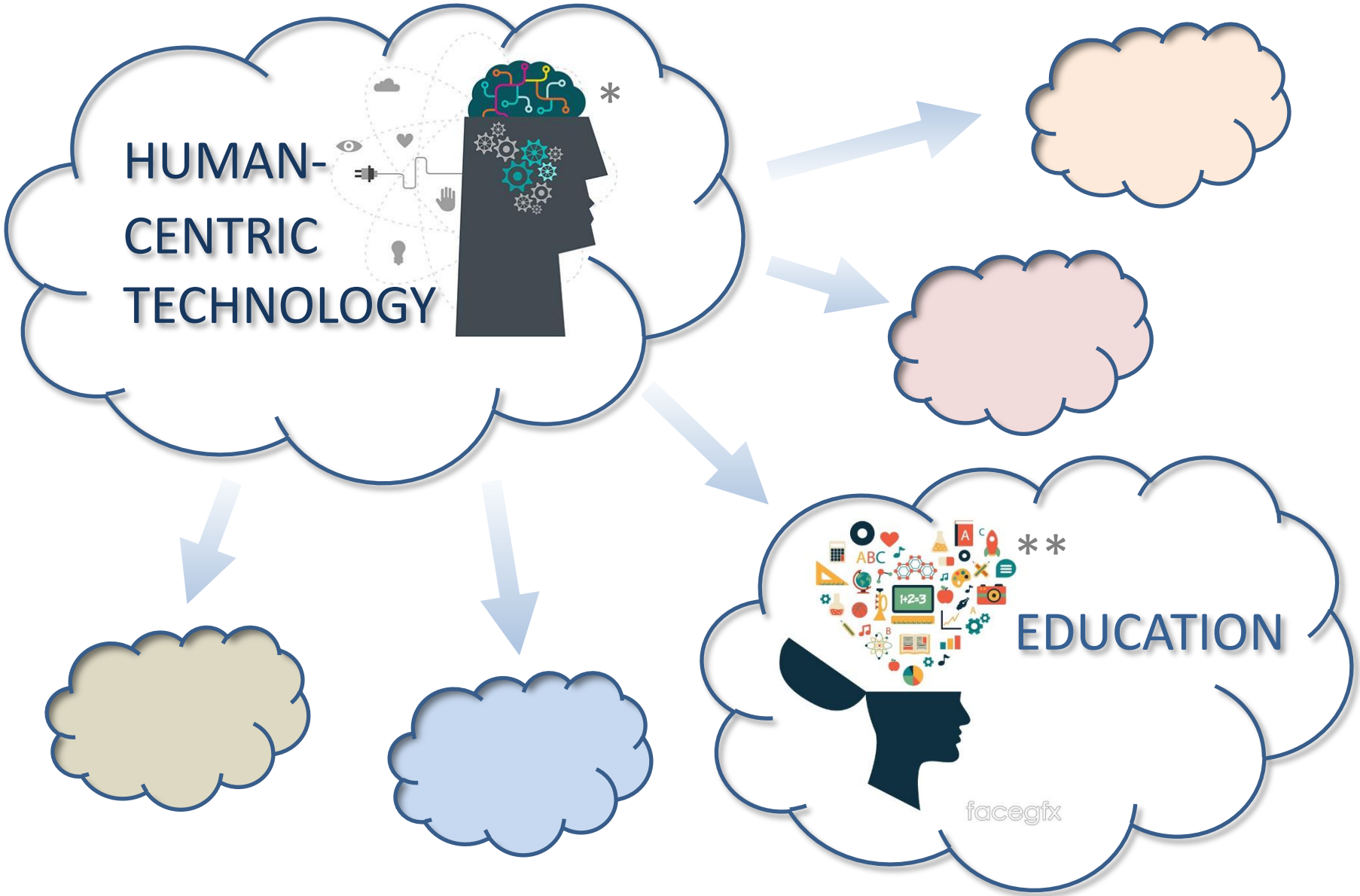


What computers can do

What users can do

Universal usability

* Shneiderman B. Leonardo's laptop: human needs and the new computing technologies. Mit Press; 2003.



* http://icc.mtu.edu/hcc/ ** http://gfx9.com

Bridge a Methodology Gap in Software Education

Attention to important particularities of software development process with respect to a software development course

Software changeability

- Much different from products of engineering

Software as a community product

- Contributing to open-source solutions requires specific skills and abilities

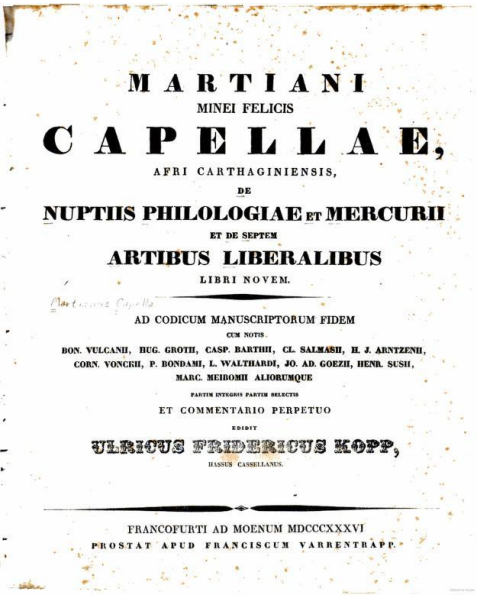
Many interdisciplinary activities

- Students have to get programming skills, but also to learn how to communicate with stakeholders, and how to cooperate in multidisciplinary teams

Programming is close to language study

- A software problem may have a variety of acceptable solutions

To the Context of Liberal Arts


MARTIANI
MINEI FELICIS
CAPELLAE,
AFRI CARTHAGINIENSIS,
DE
NUPTIIS PHILOLOGIAE ET MERCURII
ET DE SEPTEM
ARTIBUS LIBERALIBUS
LIBRI NOVEM.
AD CODICUM MANUSCRIPTORUM FIDEM
CUM NOTIS
BON. VULCANII, HUG. GROTII, CASP. BARTHII, CL. SALMASII, H. J. ARNTZENII,
CORN. VONCKII, P. BONDAMI, L. WALTHARDI, JO. AD. GOEZII, HENR. SUSII,
MARC. MEIBOMII ALIORUMQUE
PARTIM INTEGRIS PARTIM SELECTIS
ET COMMENTARIO PERPETUO
EDIDIT
ULRICUS FRIDERICUS KOPP,
HASSUS CASSELLANUS.
FRANCOFURTI AD MOENUM MDCCCXXXVI
PROSTAT APUD FRANCISCUM VARRENTRAPP.

- Arithmetic
- Geometry
- Astronomy
- Logic
- Grammar
- Rhetoric
- Music

Computer Science

Ultimate of Liberal Arts?

- Computer science draws upon perspectives from many disciplines and has a symbiotic relationship with the liberal arts disciplines, so it might be considered the ultimate of them*

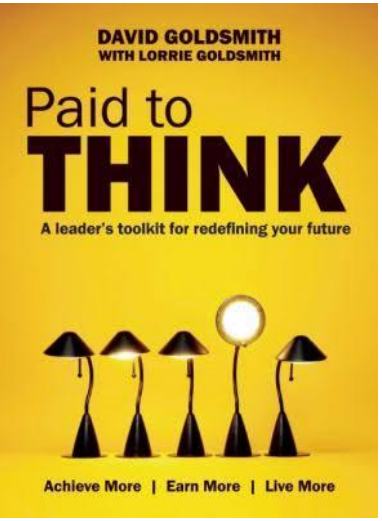
Image: Extracted from Google image search output for “computer science”

* H.M. Walker and C. Kelemen, “Computer science and the liberal arts: a philosophical examination,” ACM Transactions on Computing Education (TOCE), Mar 1, 2010, vol. 10, no. 1, pp.2:1–2:10.

Ultimate of Liberal Arts?

as well as perhaps of any kind of good education ☺

- A primary goal of *computer science education* is to develop students' abilities to ***think effectively***,



and it is no less important than developing *specific skills* required for the successful career and professional growth:

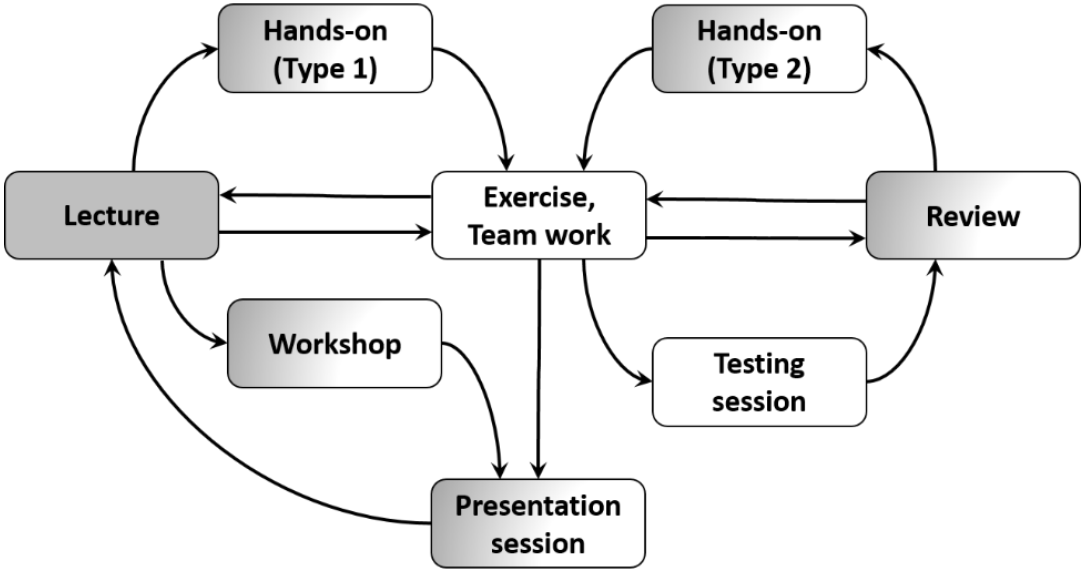
soft skills

*Excellent experts are **"paid to think"****

* David Goldsmith. 2012. Paid to Think: A Leader's Toolkit for Redefining Your Future. BenBella Books, Inc.

Use Case:
Learning Activities in a Programming Course

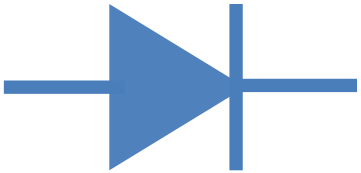
- Practices and lessons learned after teaching the connected undergraduate courses “Introduction to Programming” and “C Programming” in the University of Aizu (128 class hours in total)

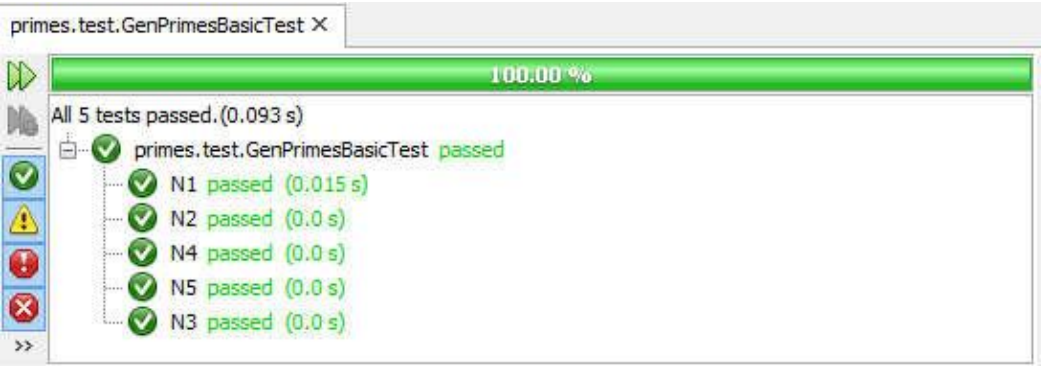


- Diversity of activity forms
  - How computer science can learn from teaching forms and practices which exists in fine arts

Lecture Organizational Drawbacks

- Often
  - Lecture is a signal to students that one-way information transmission will predominate





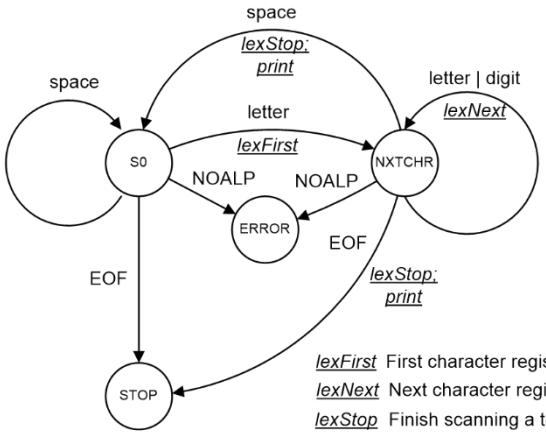
- General principles are delivered together with a pack of nicely working examples

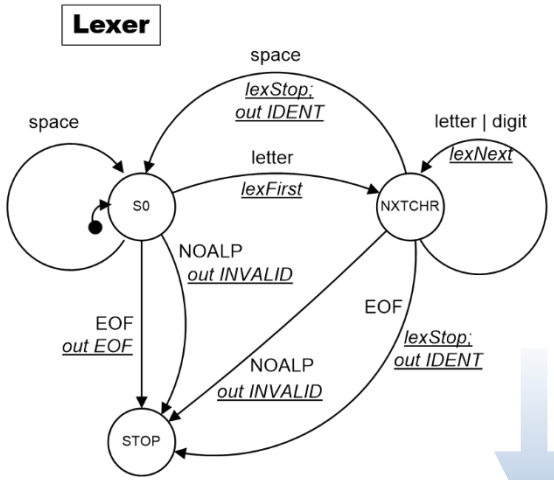
- Not much experiments with live examples with demonstrating the process with possible design errors, execution bugs, as well as with code and requirement modifications

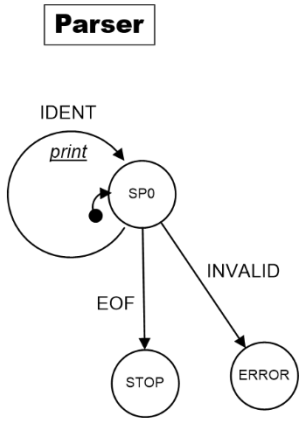
Lab/Exercise Session: Limited Cooperation

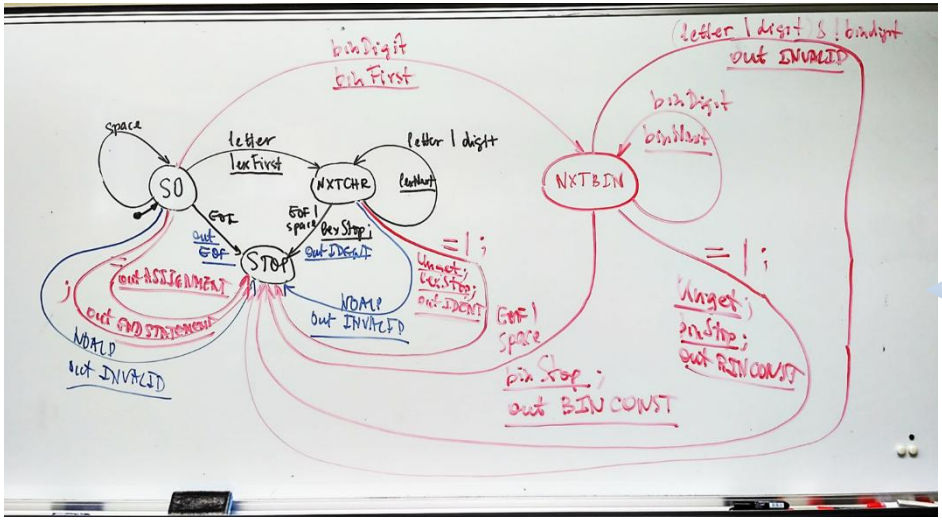
- Common drawbacks
  - Students work mostly independently
  - Anti-collaboration pattern: **any kind of assistance (except from instructors) is unauthorized**
- Different kinds of projects required:
  - Projects are assigned and performed individually
  - Tasks are personalized, however, collaboration is not forbidden
  - Teamwork (including pairs and mixing teams)

Use Case: Hands-On Session Example









Exercise: Let's Extend Our Model

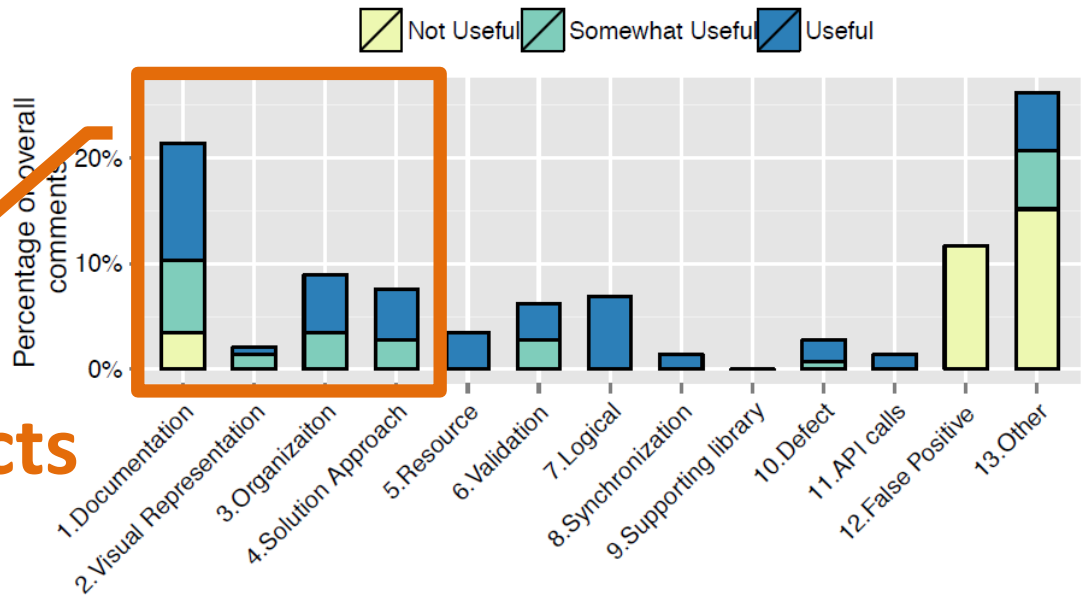
- Now our language supports three types of tokens:
  - Identifiers
  - Assignment =
  - End-of-expression ;
  - Binary values 100011101

  Lexer's responsibility
- Input file contains expressions like the following:
  - value = 100011101;

  Parser's responsibility
- The problem
  - Revise the state diagrams
  - Implement the revised finite state machines
    - Lexer recognizes tokens
    - Parser follows syntax rules
  - In process of program execution all the binary values should be printed as hexadecimal values

Code Review

- “Human-based” quality assurance practice
- Extends learning process and enforcing practicing “soft skills”
- Around 40% of useful review comments are related to **evolvability defects**



* Bosu A., Greiler M., and Bird C., “Characteristics of useful code reviews: An empirical study at Microsoft,” 12th Working Conference on Mining Software Repositories, 2015

Code Readability

- Buse & Weimer: ***Descriptive model of software readability****
- Based on simple features that can be extracted from programs automatically
  - Descriptive model can be used to predict human readability judgments for existing software
  - Very useful for academic purposes: students have to learn how to produce readable solutions

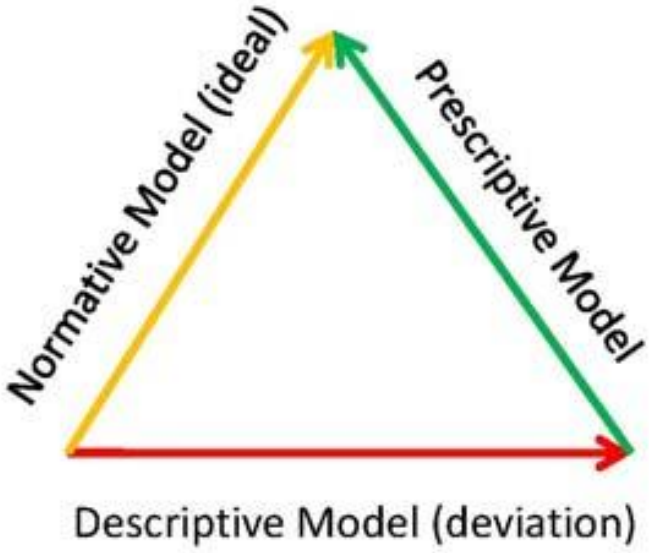


Image: https://www.researchgate.net/publication/263030211

* Buse R. and Weimer W.R., "Learning a metric for code readability," IEEE Transactions on Software Engineering 36, 4, 2010.

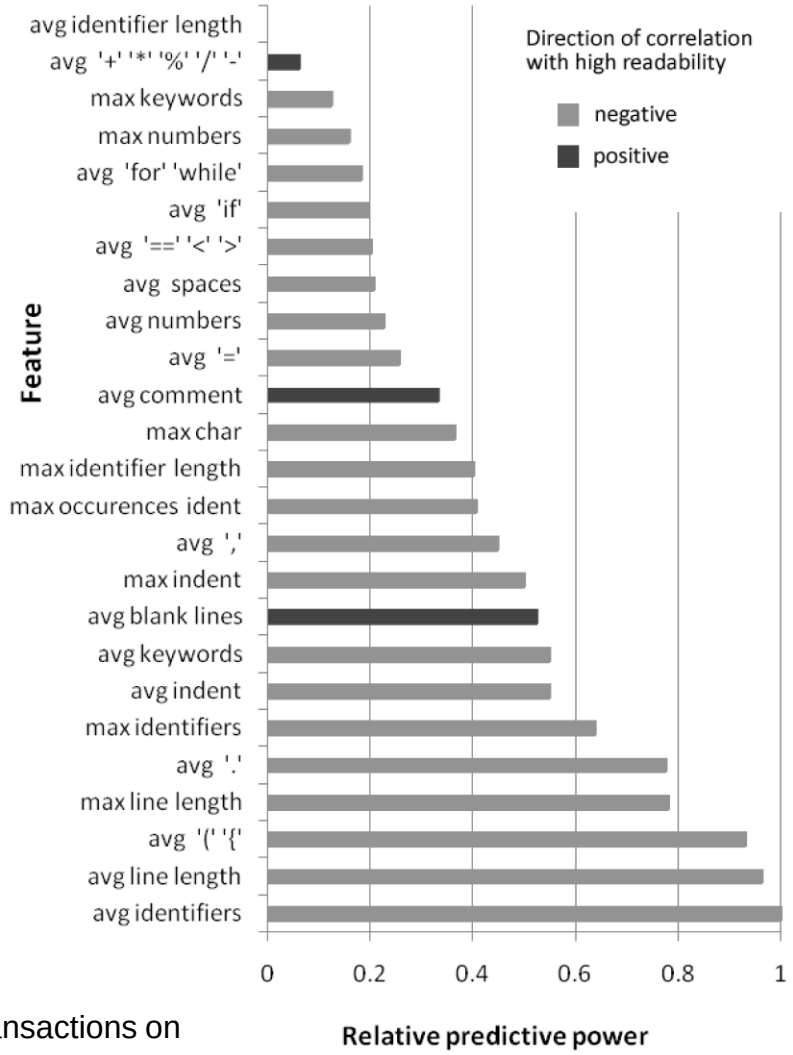
Readability Features: Example

Strong influence on readability

- Naming conventions
- Empty lines
- Number of identifiers and characters per line
- Specific indentation scheme

Moderate predictive readability power

- Long identifier names
  - They are not directly connected to a concept of self-documented code
- Comments
  - While comments can enhance readability, they are typically used in code segments that started out less readable



* Buse R. and Weimer W.R., "Learning a metric for code readability," IEEE Transactions on Software Engineering 36, 4, 2010.

Workshop “The Form inside the Work”

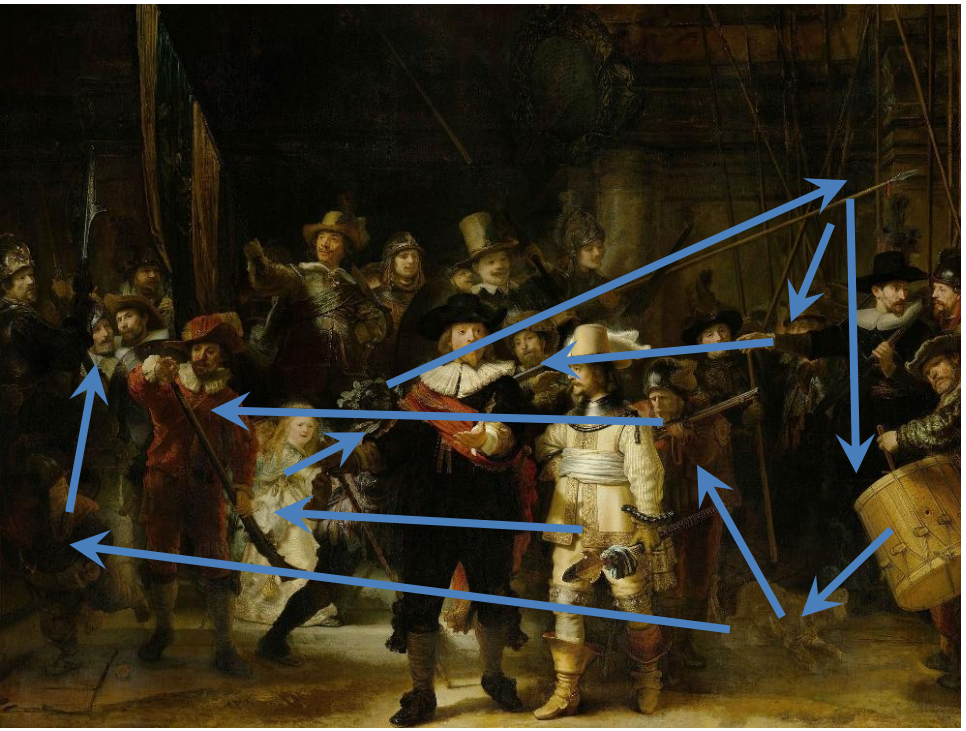
L’oeil suit les chemins qui lui ont été ménagés dans l’oeuvre

Paul Klee, quoted in George Perec’s *La Vie, mode d’emploi* (1978)

Workshop “The Form inside the Work”

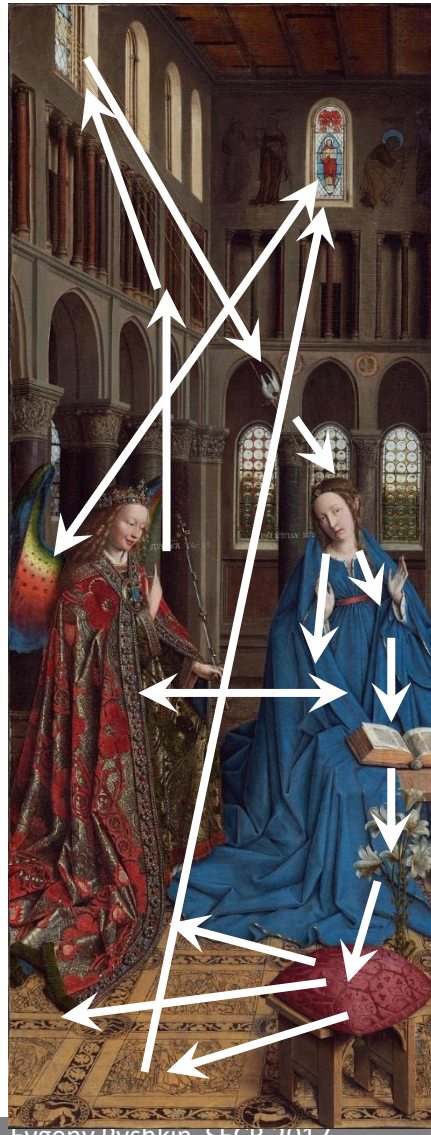
L’oeil suit les chemins qui lui ont été ménagés dans l’oeuvre

Paul Klee, quoted in George Perec’s *La Vie, mode d’emploi* (1978)



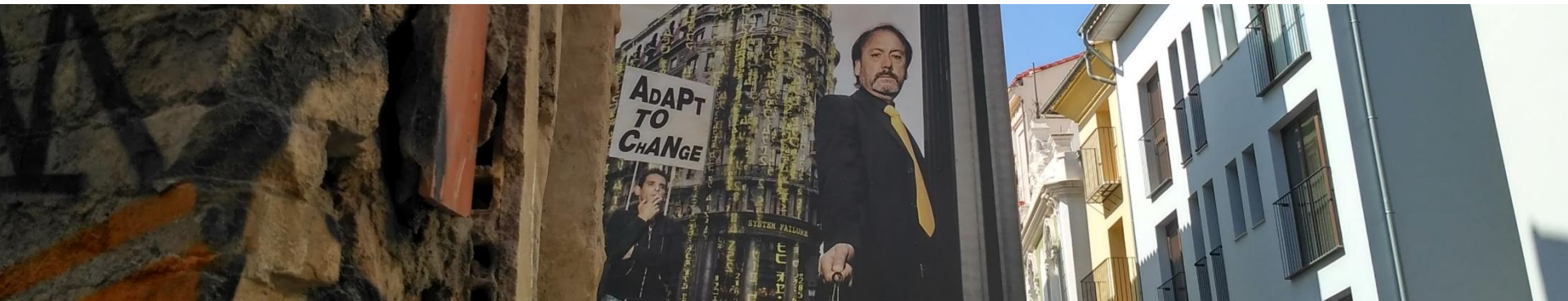
Unless you write a Chinese character in the right order of brush strokes, it would never look beautiful!

道



(PERHAPS OBVIOUS) CONCLUSIONS

- Considering CS and software disciplines within the context of liberal arts is connected to significant changes in the teaching and learning models
  - We anticipate more than only professional developers' skills from our students
    - They have to be able to work in a collaborative environment
    - Significance of organizational learning models favoring **public display, teamwork and professional discussion** significantly increases
- It is extremely important to find ways to create a collaboration environment where students can actively participate in the **co-learning** process together with their more experienced colleagues



SECR
Software Engineering Conference Russia
St. Petersburg October 20 – 22 2017

LIBERAL ARTS IN A DIGITALLY TRANSFORMED WORLD: REVISITING A CASE OF SOFTWARE DEVELOPMENT EDUCATION

Evgeny Pyshkin

University of Aizu

会津大学