Evolving Software: A Fusion of the Developer and Evolutionary Algorithms

Chris Simons

Department of Computer Science and Creative Technologies

University of the West of England

Bristol BS16 1QY United Kingdom

@chrislsimons

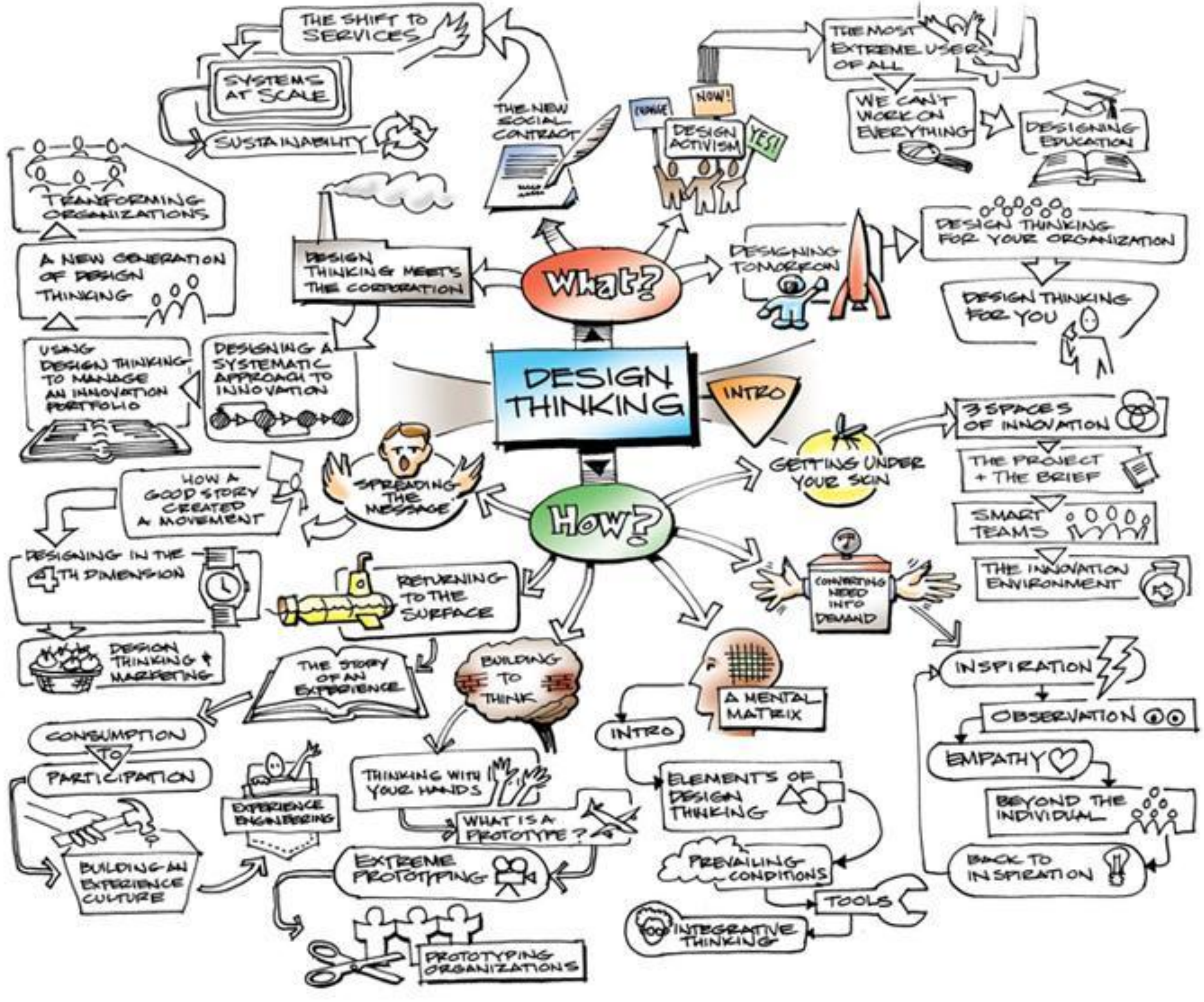
chris.simons@uwe.ac.uk

www.cems.uwe.ac.uk/~clsimons

ACCU Conference for 2015
Bristol, UK
21 – 15 April 2015

Agenda

- Motivation - Software Design Evolution
- Evolutionary Algorithms (EAs)
- Fitness measures for evolving software
  - Breakout
- A fusion of software engineer and computer
  - Breakout


DESIGN THINKING
What?
How?
INTRO
THE SHIFT TO SERVICES
SYSTEMS AT SCALE
SUSTAINABILITY
TRANSFORMING ORGANIZATIONS
A NEW GENERATION OF DESIGN THINKING
USING DESIGN THINKING TO MANAGE AN INNOVATION PORTFOLIO
DESIGNING A SYSTEMATIC APPROACH TO INNOVATION
DESIGN THINKING MEETS THE CORPORATION
THE NEW SOCIAL CONTRACT
CHANGE
NOW!
DESIGN ACTIVISM
YES!
THE MOST EXTREME USERS OF ALL
WE CAN'T WORK ON EVERYTHING
DESIGNING EDUCATION
DESIGNING TOMORROW
DESIGN THINKING FOR YOUR ORGANIZATION
DESIGN THINKING FOR YOU
3 SPACES OF INNOVATION
THE PROJECT + THE BRIEF
SMART TEAMS
THE INNOVATION ENVIRONMENT
GETTING UNDER YOUR SKIN
CONVERTING NEED INTO DEMAND
INSPIRATION
OBSERVATION
EMPATHY
BEYOND THE INDIVIDUAL
BACK TO INSPIRATION
A MENTAL MATRIX
INTRO
ELEMENTS OF DESIGN THINKING
PREVAILING CONDITIONS
TOOLS
INTEGRATIVE THINKING
SPREADING THE MESSAGE
HOW A GOOD STORY CREATED A MOVEMENT
DESIGNING IN THE 4TH DIMENSION
DESIGN THINKING & MARKETING
RETURNING TO THE SURFACE
THE STORY OF AN EXPERIENCE
BUILDING TO THINK
CONSUMPTION TO PARTICIPATION
BUILDING AN EXPERIENCE CULTURE
EXPERIENCE ENGINEERING
THINKING WITH YOUR HANDS
WHAT IS A PROTOTYPE?
EXTREME PROTOTYPING
PROTOTYPING ORGANIZATIONS

Brown, T. (2009) *Change by Design*, New York, Harper Collins.

Software design is ***complex…***

… and an ***human*** activity.

Special Issue Studying Professional Software Design *Design Studies*, vol. 31, no. 6, pp. 533 – 662, 2010.

Representing structure in a software system design

Michael Jackson

Design requirements, epistemic uncertainty and solution development strategies In software design

Linden J. Ball, Balder Onarheim, Bo T. Christensen

Ideas, subjects, and cycles as lenses for understanding the software design process

Alex Baker, André van der Hoek

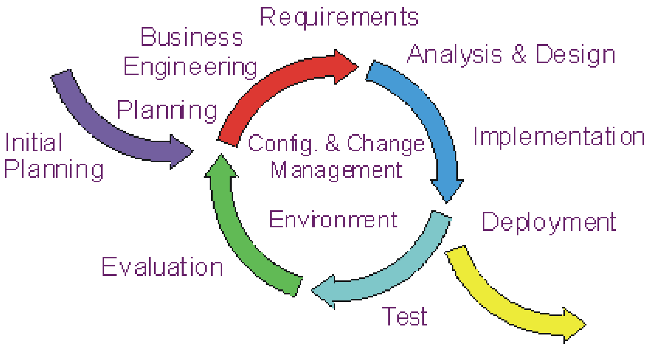
What makes software design effective?

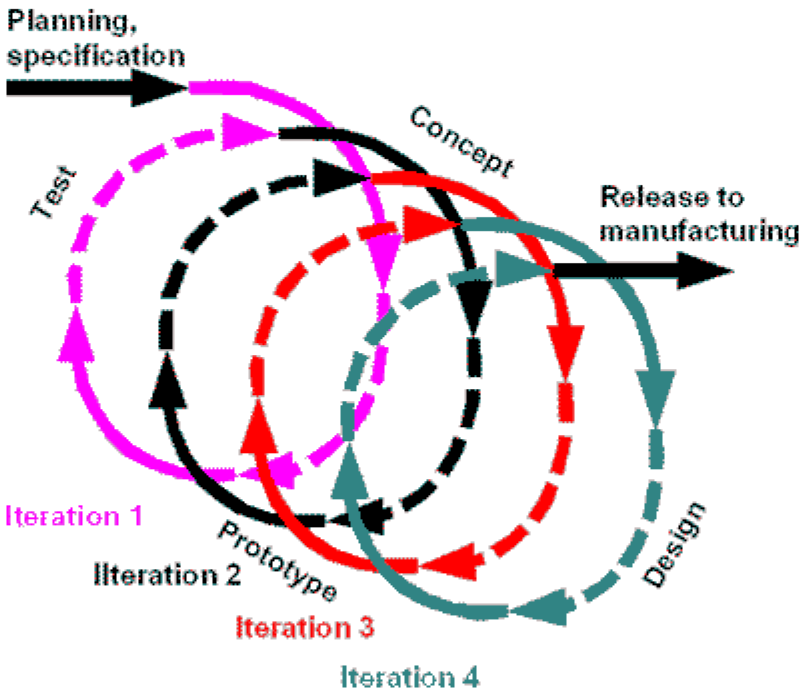
Antony Tang, Aldeida Aleti, Janet Burge, Hans van Vliet

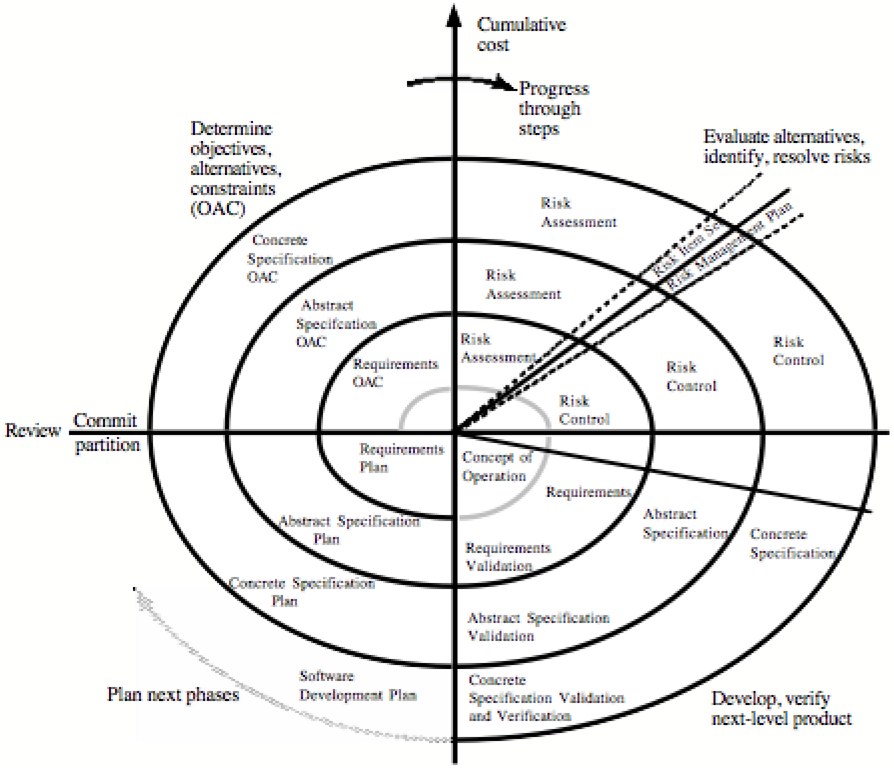
Accessing decision-making in software design

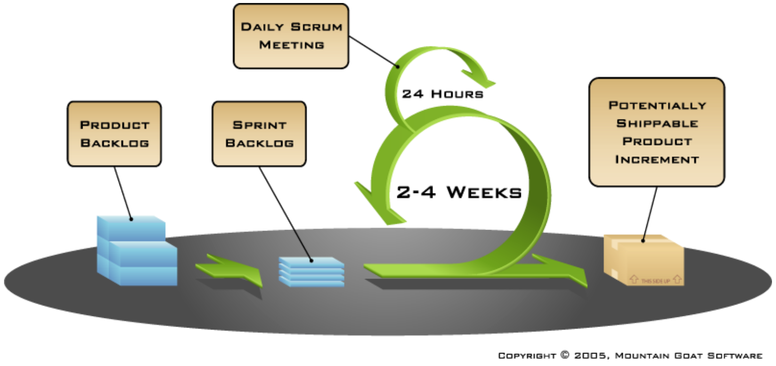
Henri Christiaans, Rita Assoreira Almendra

Agile: Iterative & Incremental









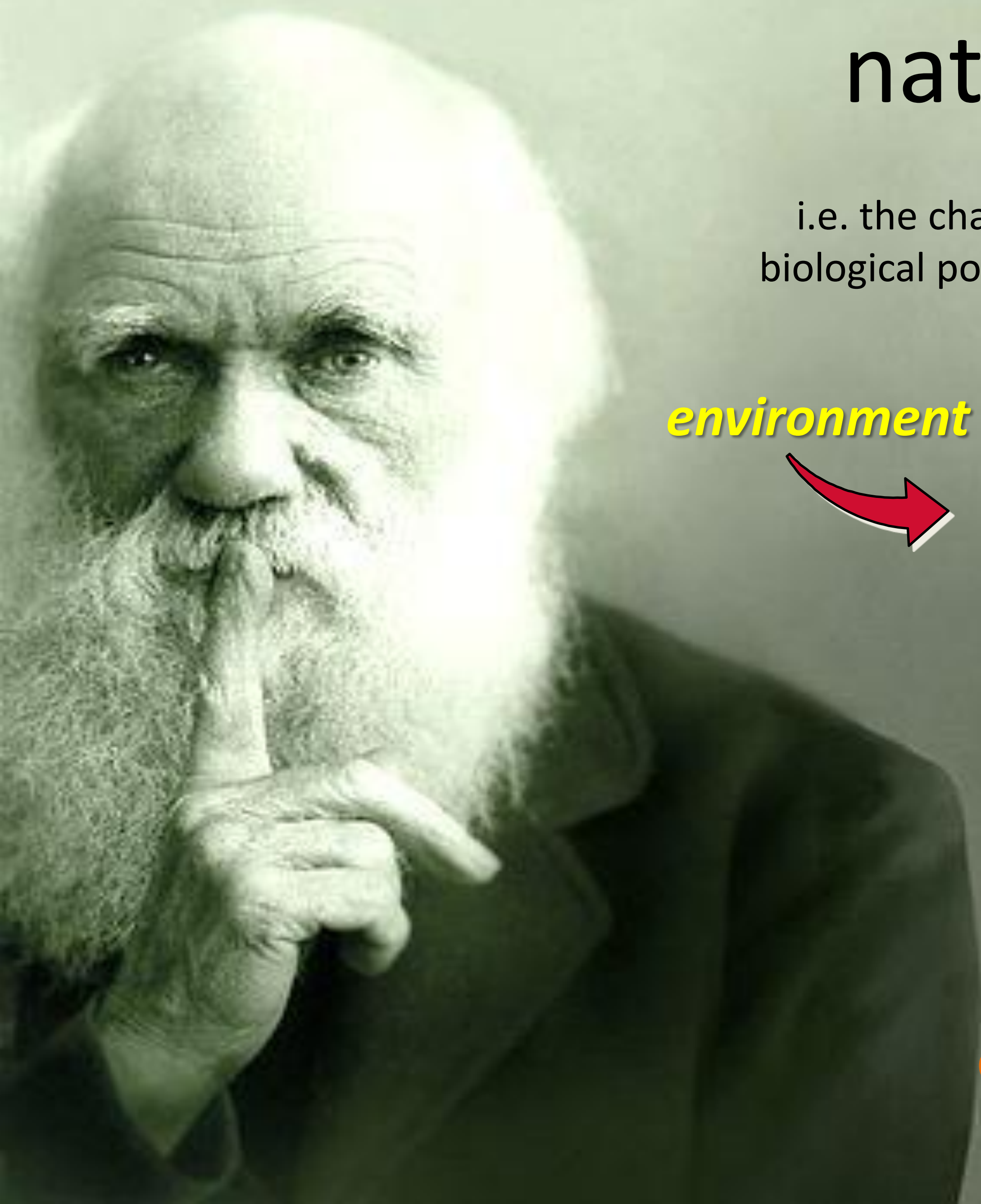
With the rise of agile methodologies, is software design dead?

http://martinfowler.com/articles/designDead.html (2004)

“For many that come briefly into contact with Extreme Programming, it seems that XP calls for the death of software design. Not just is much design activity ridiculed as "Big Up Front Design", but such design techniques as the UML, flexible frameworks, and even patterns are de-emphasized or downright ignored….”

“…In fact XP involves a lot of design, but does it in a different way than established software processes. XP has rejuvenated the notion of evolutionary design with practices that allow evolution to become a viable design strategy. It also provides new challenges and skills as designers need to learn how to do a simple design, how to use refactoring to keep a design clean, and how to use patterns in an evolutionary style.”

www.flickr.com/photos/armincifuentes/

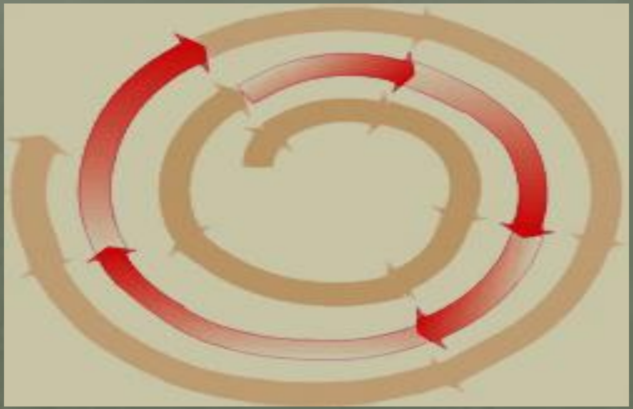


natural evolution

i.e. the change in the inherited characteristics of biological populations over successive generations.

environment

Selection of fittest individuals



sexual reproduction for
diversity and population change

Evolutionary Algorithms…

Not new…

- Alan Turin (1952)
    - “Computing Machinery and Intelligence” in *Mind*
    - hints at a *“…genetical programming...”*
- Alex Fraser (1957)
    - Computational simulation of natural evolution
- Fogel *et al.* (1966)
    - *Evolutionary programming* (finite state machines)
- Rechenburg (1973)
    - *Evolutionary Strategies*
- Holland (1975)
    - *Genetic Algorithms*
- Kosa (1992)
    - *Genetic Programming*

And many more

…computational evolution

Representation of an “individual” solution

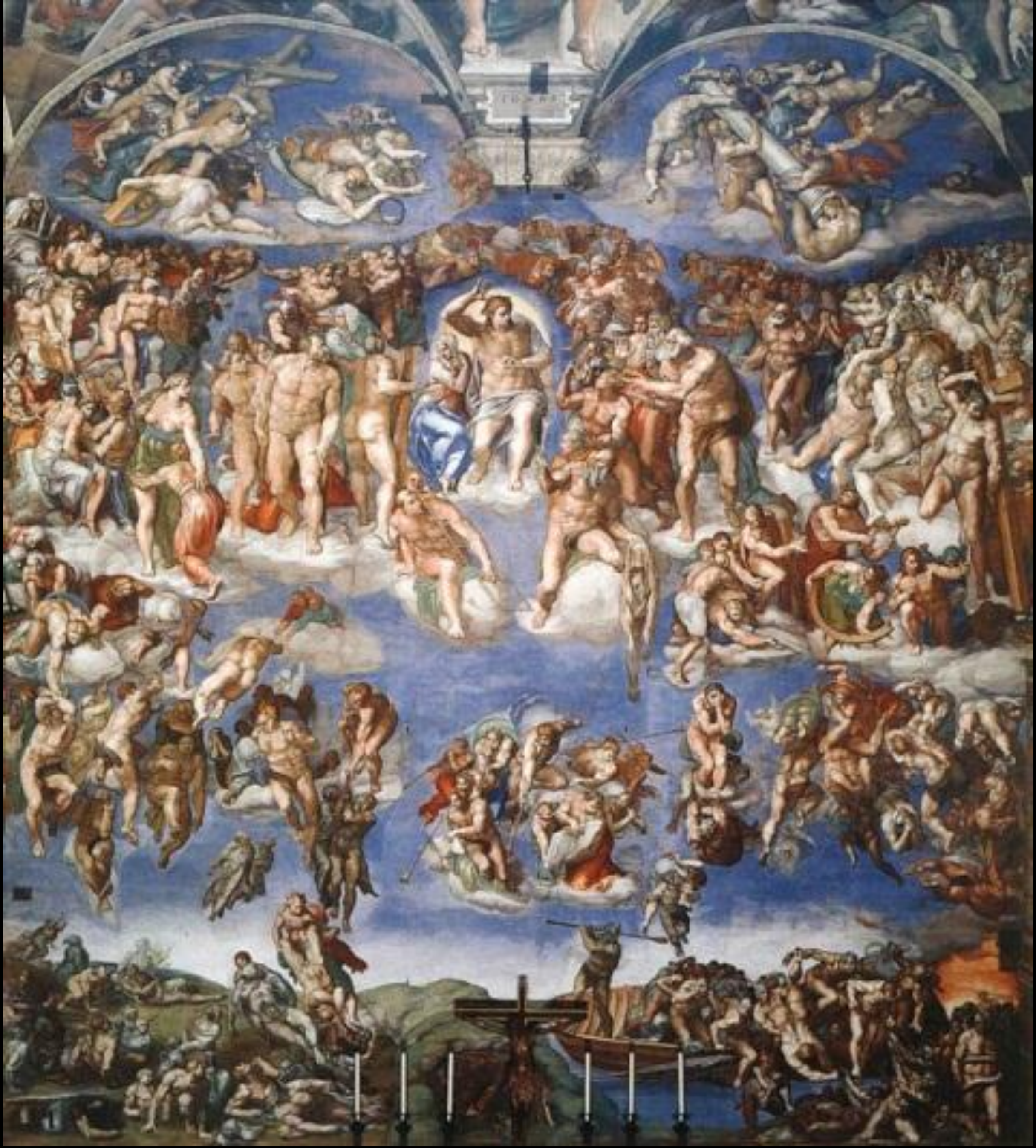
e.g. models, trees, arrays etc. etc.

```
initialise population at random
while( not done )
  evaluate each individual
  select parents
  recombine pairs of parents
  mutate new candidate individuals
  select candidates for next generation
end while
```

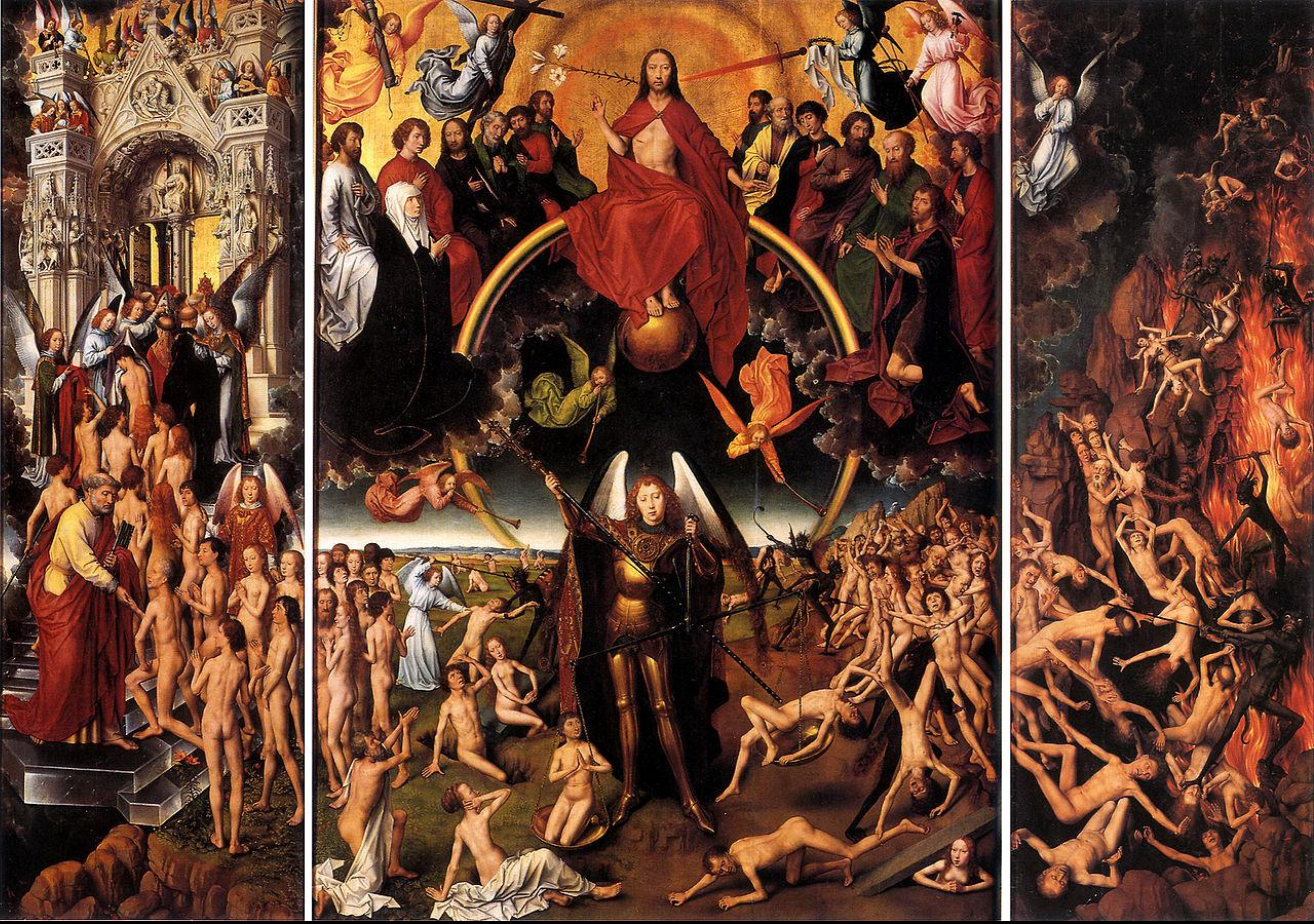
Eiben, A.E., Smith, J.E. (2003) *Introduction to Evolutionary Computing*, Springer.

but what does ‘evaluation’ mean for a software design?

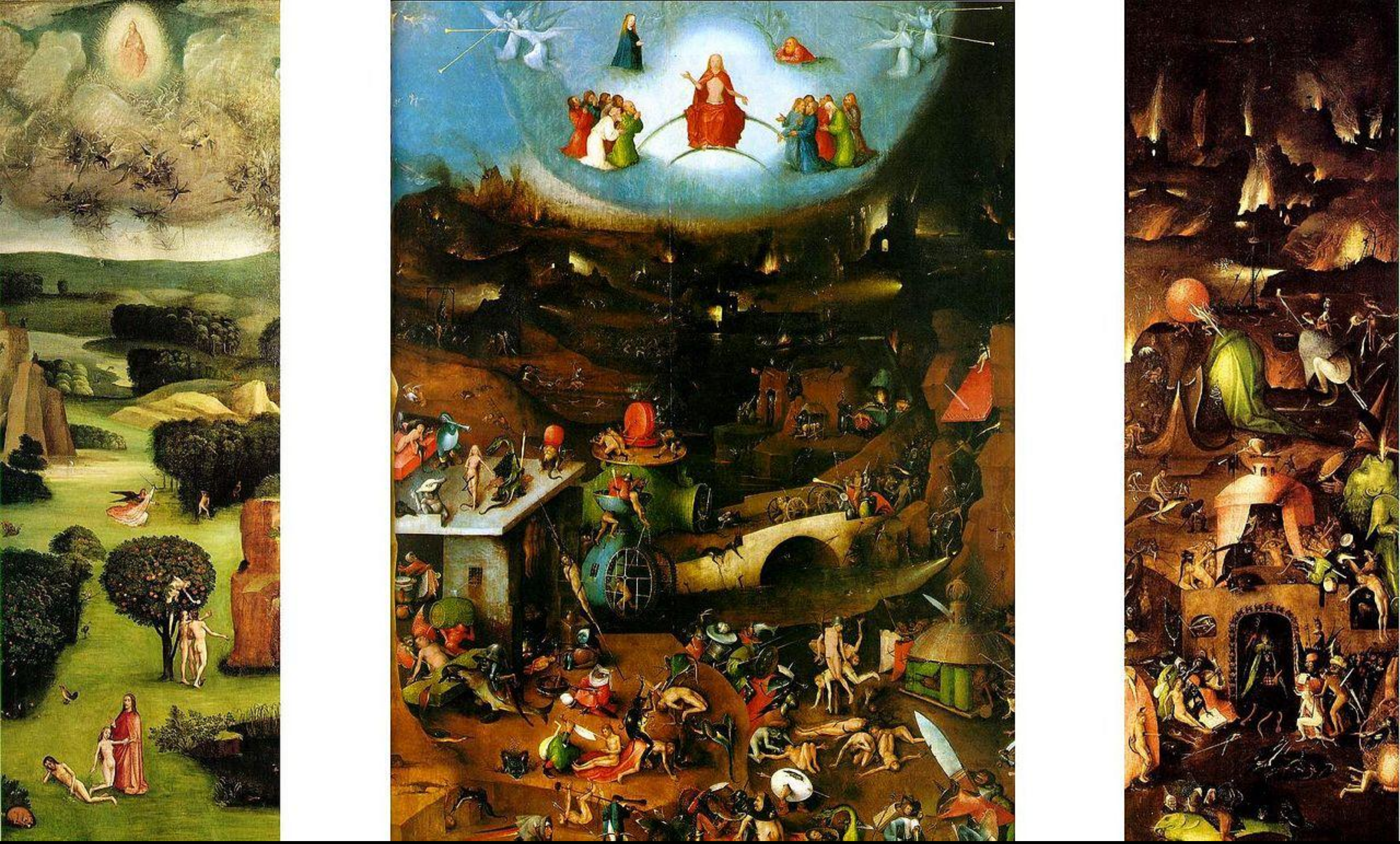
…given that software design is complex and intensely human-centred



http://en.wikipedia.org/wiki/Judgement_day#mediaviewer/File:Jugement_dernier.jpg



http://en.wikipedia.org/wiki/Judgement_day#mediaviewer/File:Das_J%C3%BCngste_Gericht_(Memling).jpg



http://en.wikipedia.org/wiki/The_Last_Judgment_(Bosch_triptych)#mediaviewer/File:Last_judgement_Bosch.jpg

ORIGINAL MOTION PICTURE SOUNDTRACK
SCHWARZENEGGER
TERMINATOR 2
JUDGMENT DAY
MUSIC BY BRAD FIEDEL

http://whoniversefanon.wikia.com/wiki/Doctor_who:_Judgement_Day

In nature - fitness is sexy?

"the significance of symmetry was only made clear with the discovery that stress and disease make it harder for an individual to develop a perfectly symmetric body. Small differences on either side of an imaginary mid-plane therefore betray genetic quality, and potential mates use this to gauge each other's desirability. Put simply, symmetry is sexy".

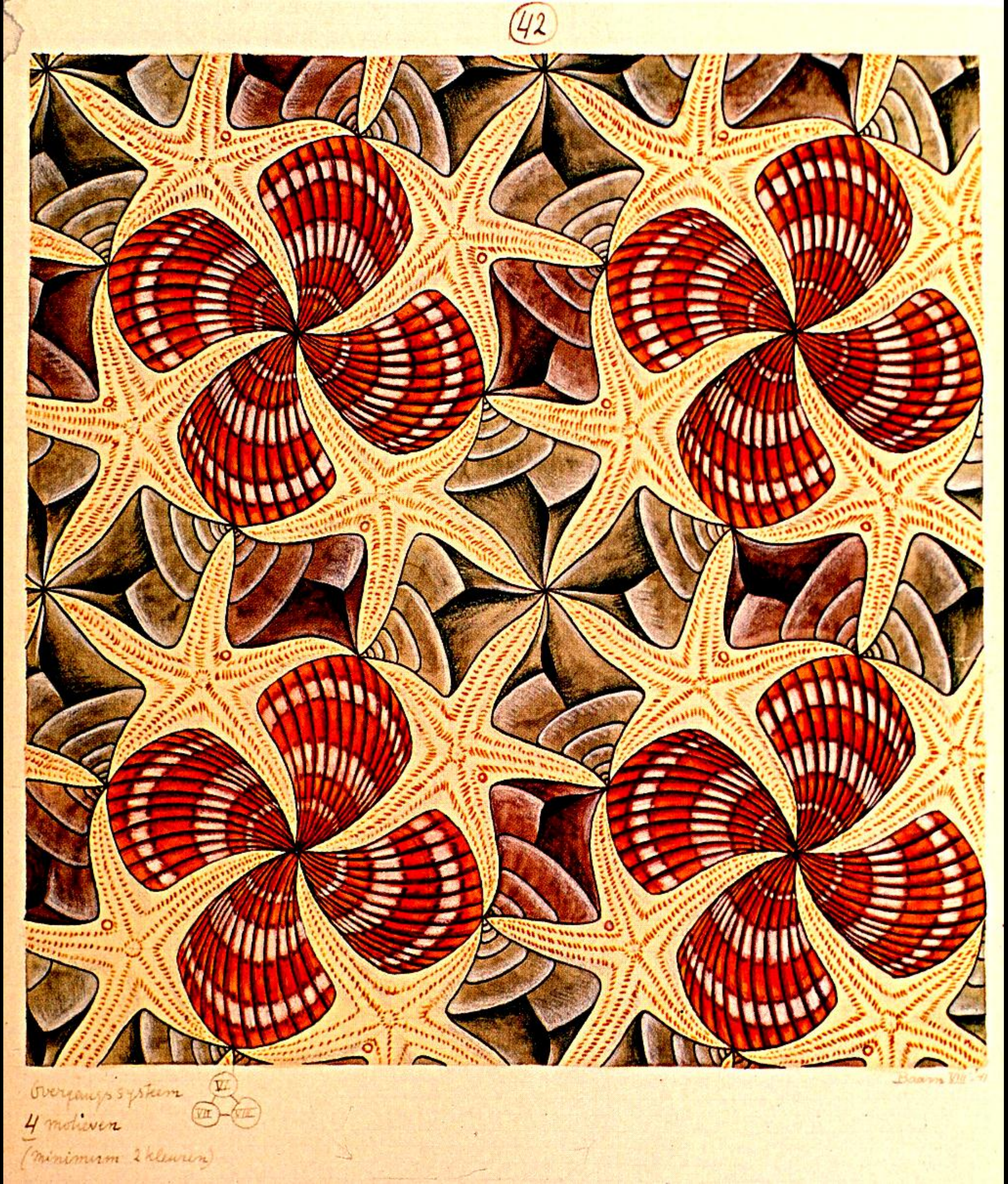
Schilthuizen, M., "Lopsided Love", *New Scientist,* 18 June 2011, pp. 42-45.

Symmetrical fitness in art and jewellery?

Fitness in tools too?

Symmetrical fitness still with us today?

Schattschneider, D. (2004) *M.C.Escher: Visions of Symmetry*, London, Thames & Hudson.


42
Overgangssysteem
4 motieven
(minimum 2 kleuren)
VI
VII
VIII
Baarn VIII-'41

“The ultimate object of design is form.”

“…every design problem begins with an effort to achieve fitness between two entities: the form in question and its context. The form is the solution to the problem, the context defines the problem.”

“…when we speak of design, the real object of the discussion is not the form alone, but the ensemble which relates to some particular division of the ensemble into form and context. ***Good fit*** is a desired property of this ensemble which relates to some particular division of the ensemble into form and context.”

? misfit is easier to recognise than fit ?

? Context == requirements? Form == software design? fitness == ??

Chapter 2, “Goodness of Fit”, in Christopher Alexander (1964) *Notes on the Synthesis of Form*, Harvard University Press.

Breakout Session (1)
Evaluation

In small groups (3/4 people), suggest possible measures to evaluate software designs.

15 minutes please

What is design “evaluation”?

Objective?

Design metrics
e.g. coupling, cohesion….

Typically quantitative

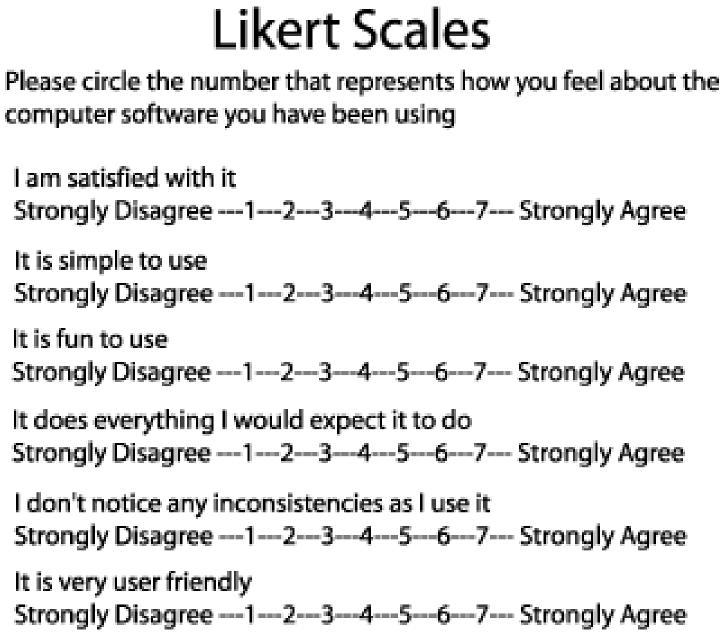
Subjective?

Value judgement
e.g. elegance, traceable, understandable….

Typically qualitative

A combination of both i.e. “multi-obsubjective”

To quantify the unquantifiable?



Likert Scales

Please circle the number that represents how you feel about the computer software you have been using

I am satisfied with it
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

It is simple to use
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

It is fun to use
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

It does everything I would expect it to do
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

I don't notice any inconsistencies as I use it
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

It is very user friendly
Strongly Disagree ---1---2---3---4---5---6---7--- Strongly Agree

(http://www.hkadesigns.co.uk/websites/msc/reme/likert.htm)

A psychometric scale commonly involved in research that employs questionnaires.

Ohsaki, M., Takagi, H., Ohya, K. (1998) An input method using discrete fitness values for interactive genetic algorithms. *Journal of Intelligent and Fuzzy Systems*, vol. 6, no. 1, pp. 131-145.

```
initialise population at random
while( not done )
  evaluate each individual
  select parents
  recombine pairs of parents
  mutate new candidate individuals
  select candidates for next generation
end while
```

Representation of an “individual” solution

e.g. models, trees, arrays etc. etc.

There are two distinct needs….

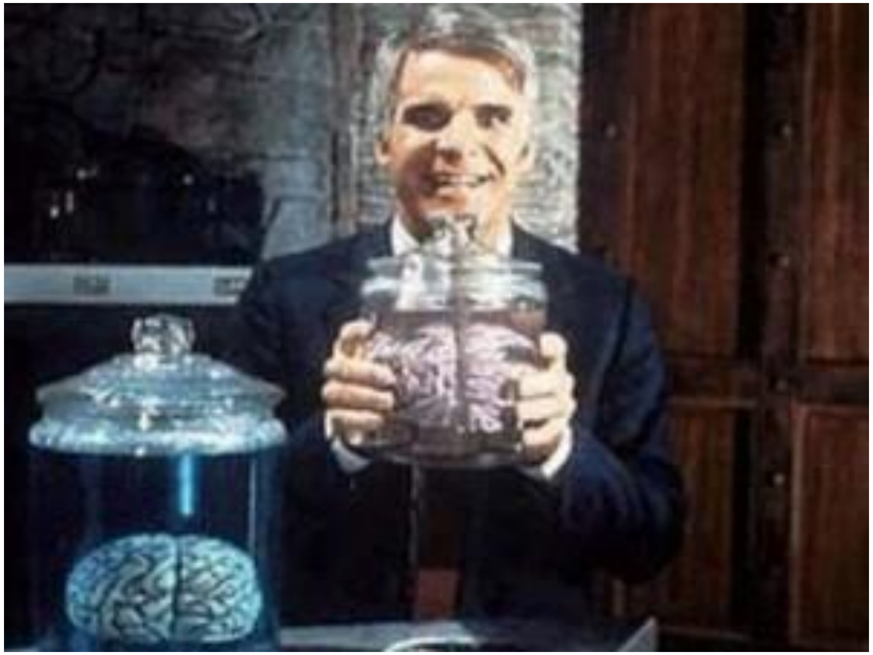
1 - Enable effective exploration and efficient (i.e. fast) search

2 - Enable effective evaluation, both objective and subjective

- graphical visualisation required?

We tried to ***replace*** people to fully automate software design and development

Didn't really work…?



-Better as a human-machine ***partnership***

-- "human-in-the-loop"

-But partnership requires agreement

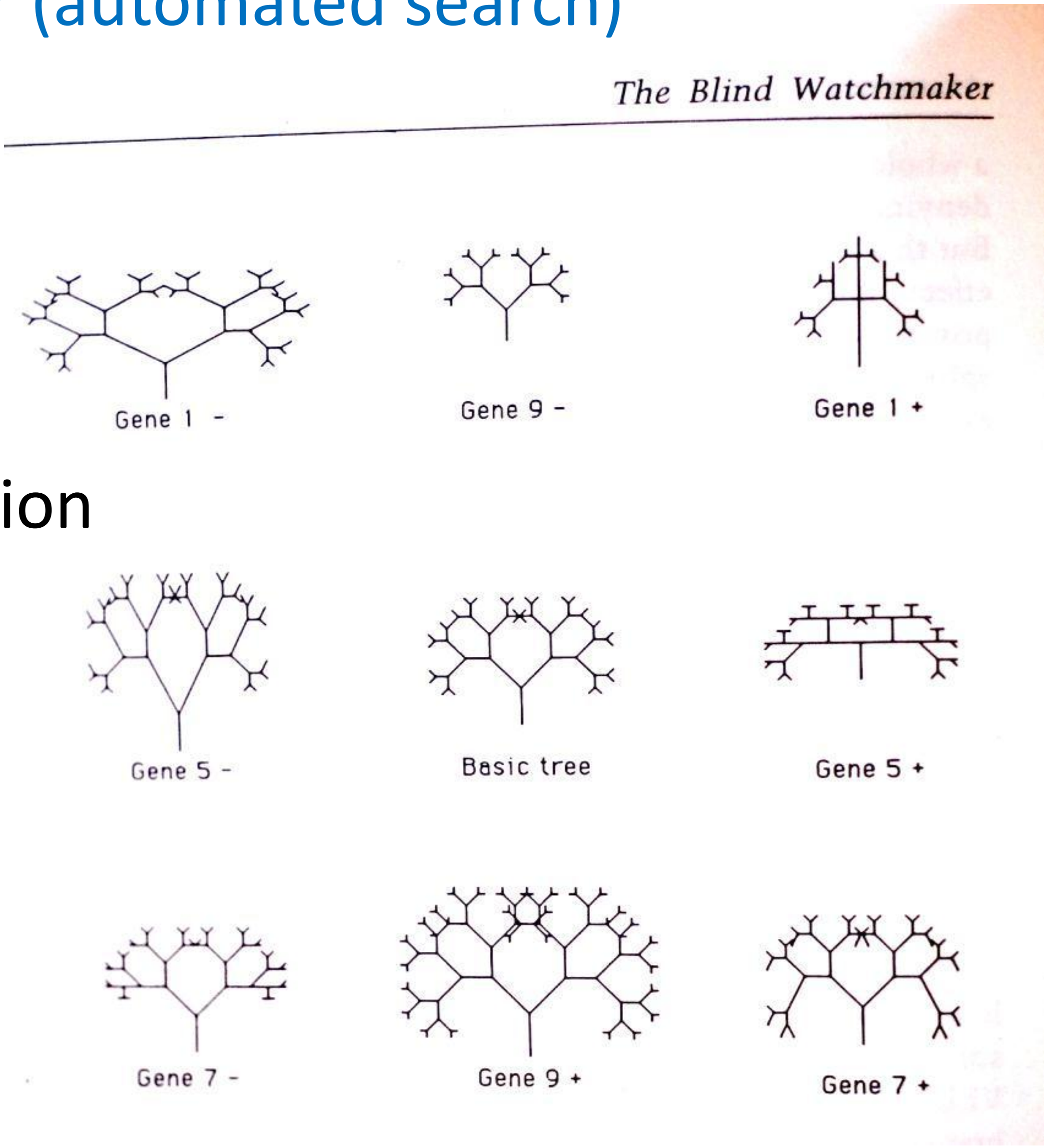
-- mutually predictable actions

-- maintain common ground

Klein, G., *et al.*, (2004) Ten Challenges for Making Automation a 'Team Player' in Joint Human-Agent Activity, *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 91-95.

The partnership becomes a ***fusion*** of software engineer (evaluation) and computer (automated search)

e.g. Dawkins 'biomorphs'



The Blind Watchmaker

Human is the fitness function

"Interactive Evolutionary Computation" (IEC)

Dawkins, R. (1986) *The Blind Watchmaker,* Penguin Books.

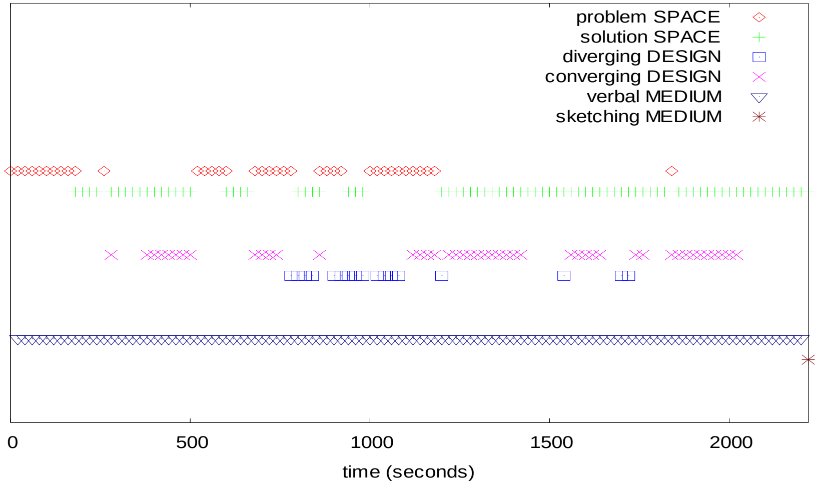
- Since ‘biomorphs’:
  - Art
  - Music
  - Image processing
  - Games
  - Industrial product design
  - Fashion Design
  - Control and robotics
  - Etc. etc. etc.

Tagaki, H. (2001) Interactive Evolutionary Computation: A Fusion of the Capabilities of EC Optimisation and Human Evaluation. *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1275-1296.

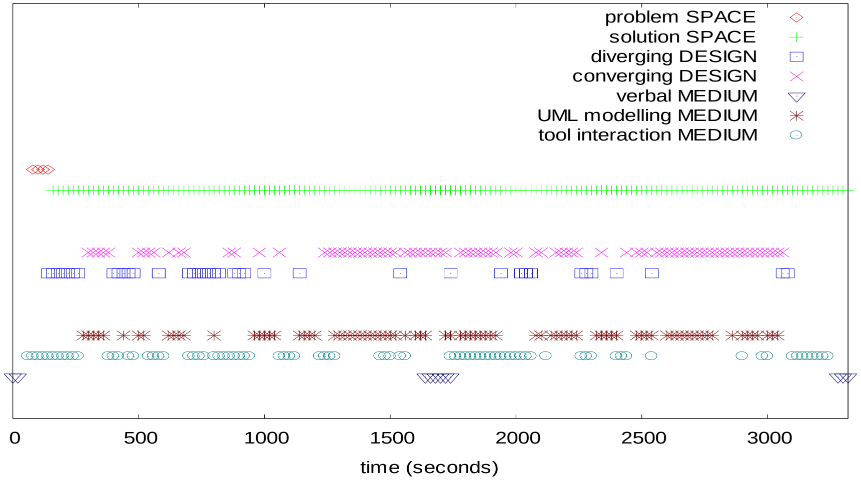
Kosorukoff, A. (2001) Human-Based Genetic Algorithm (HBGA). *Proceedings of the 2001 IEEE Int’l Conf. Systems, Man, and Cybernetics,* vol. 5, pp. 3464-3469.

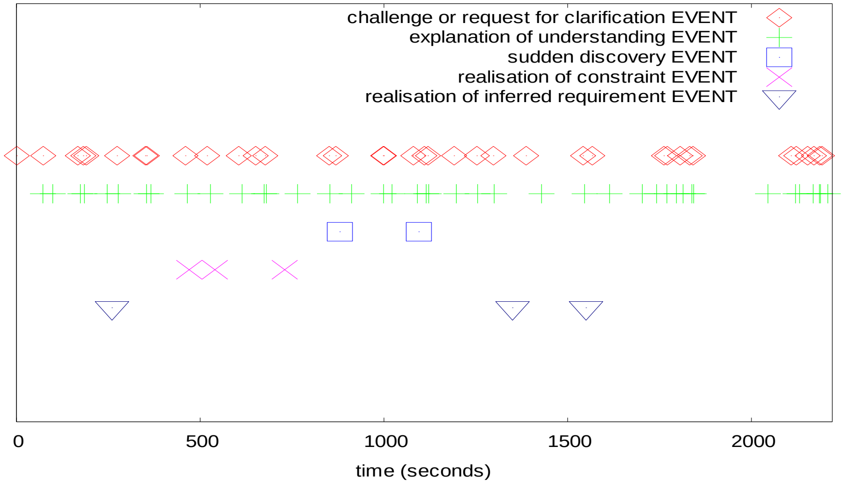
Example empirical study of interactive search of software design search space – the fusion

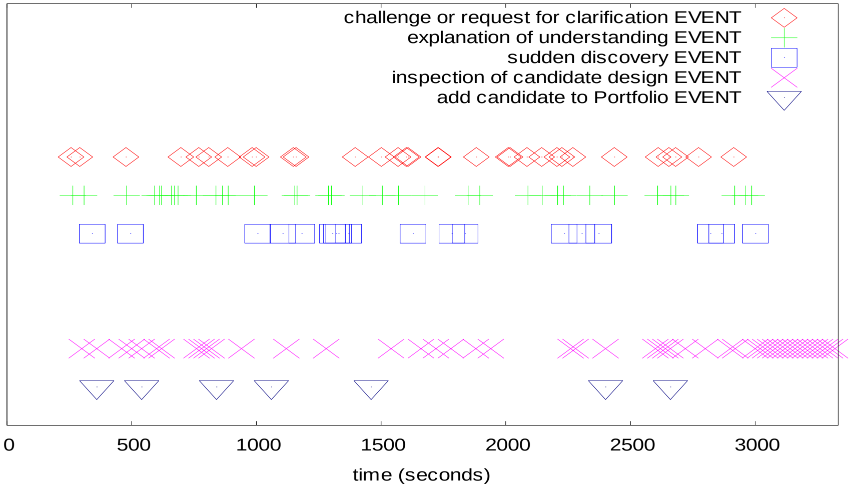
Manual design episode



Interactive design episode



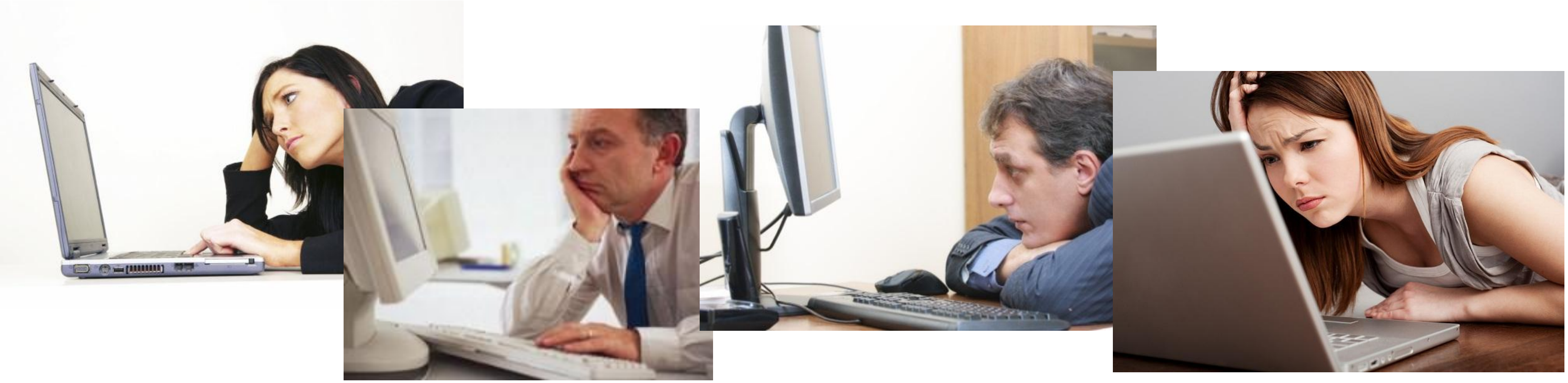




Simons, C.L., Parmee, I.C. (2009) An Empirical Investigation of Search-based Computational Support for Conceptual Software Engineering Design, in *Proceedings of the 2009 IEEE Int'l Conf. Systems, Man, and Cybernetics, (SMC '09)* , pp. 2577-2582.

Effective Interactive Search (1)

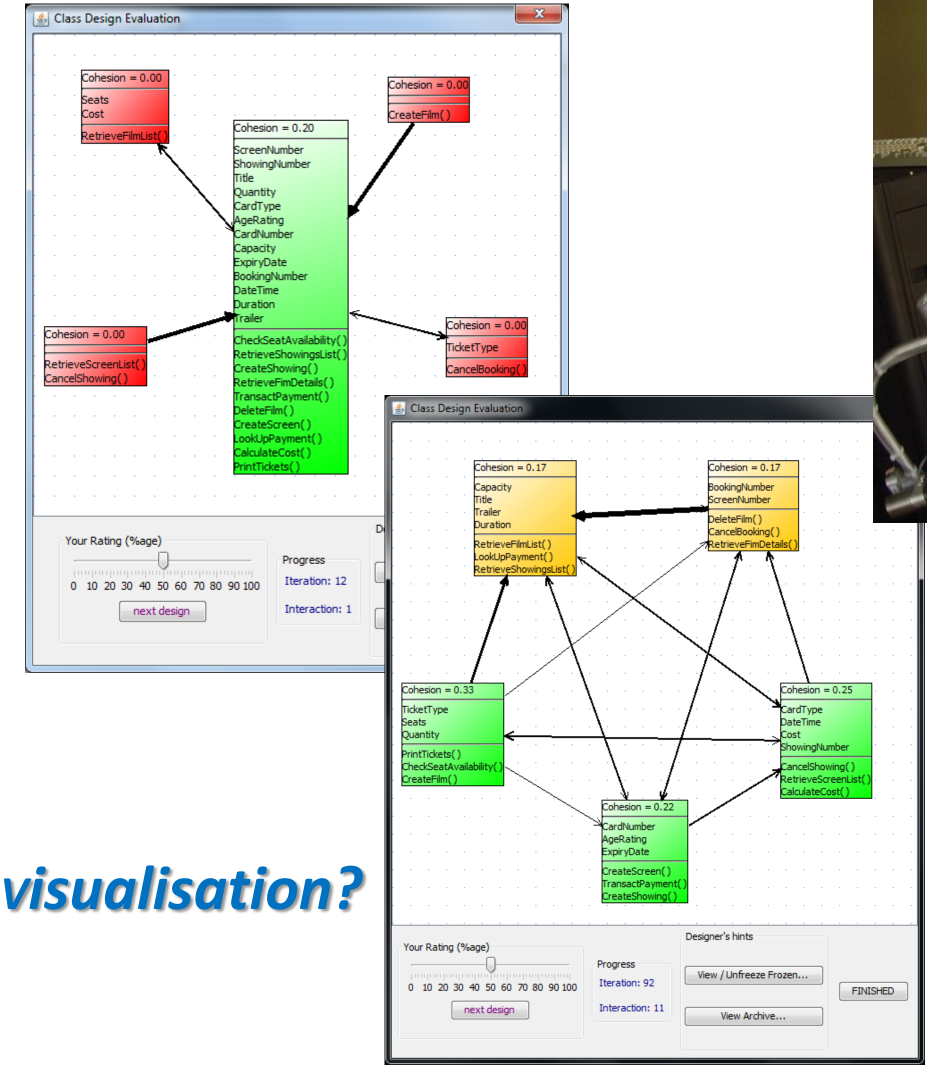
But over time, we may see a non-linearity of focus…



Risk of designer “interaction fatigue” – ***termination criteria?***

Effective Interactive Search (2)

Or more sophisticated interaction?



Your Rating (%age)

Progress

Iteration: 12

Interaction: 1

Your Rating (%age)

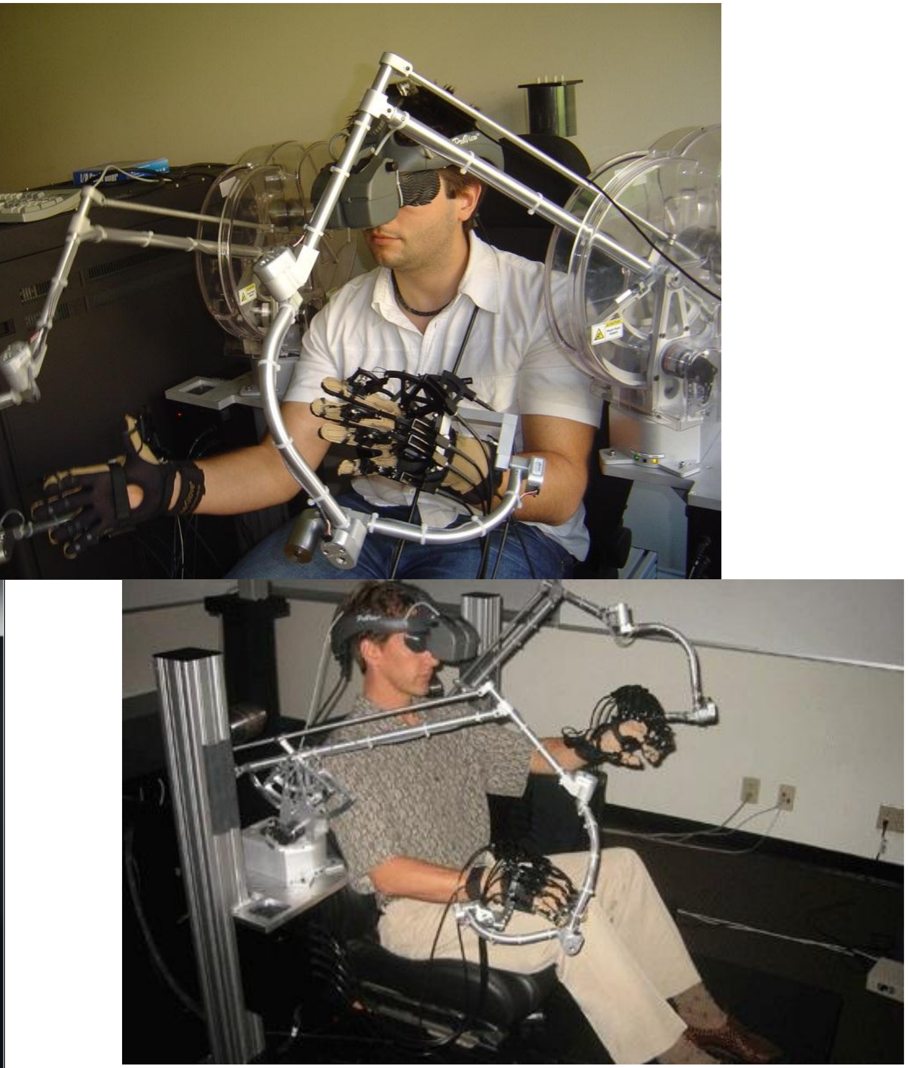
Progress

Iteration: 92

Interaction: 11

Designer's hints

visualisation?



http://teslacore.blogspot.co.uk/2005/06/epfl-haptic-workstation.html

Breakout Session (3)

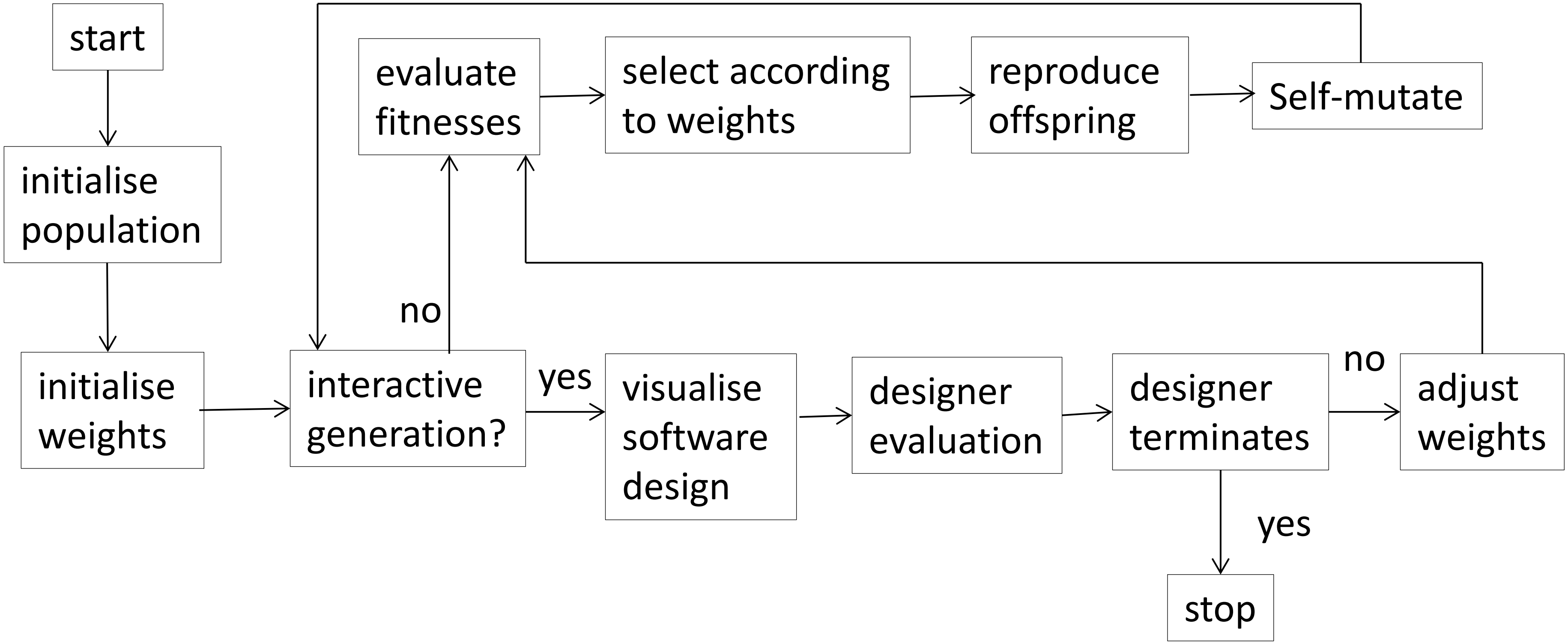
Interactive Search Approach

In small groups (3/4 people), suggest possible interactive evolutionary algorithms to to evolve software designs.

Flowcharts, pseudocode, anything that works!
15 minutes please

Example (1) Interactive Evolutionary Computing

- Object-oriented software design elegance metrics
  - E.g. Numbers among Classes (NAC)



start

initialise population

initialise weights

interactive generation?

no

evaluate fitnesses

select according to weights

reproduce offspring

Self-mutate

yes

visualise software design

designer evaluation

designer terminates

no

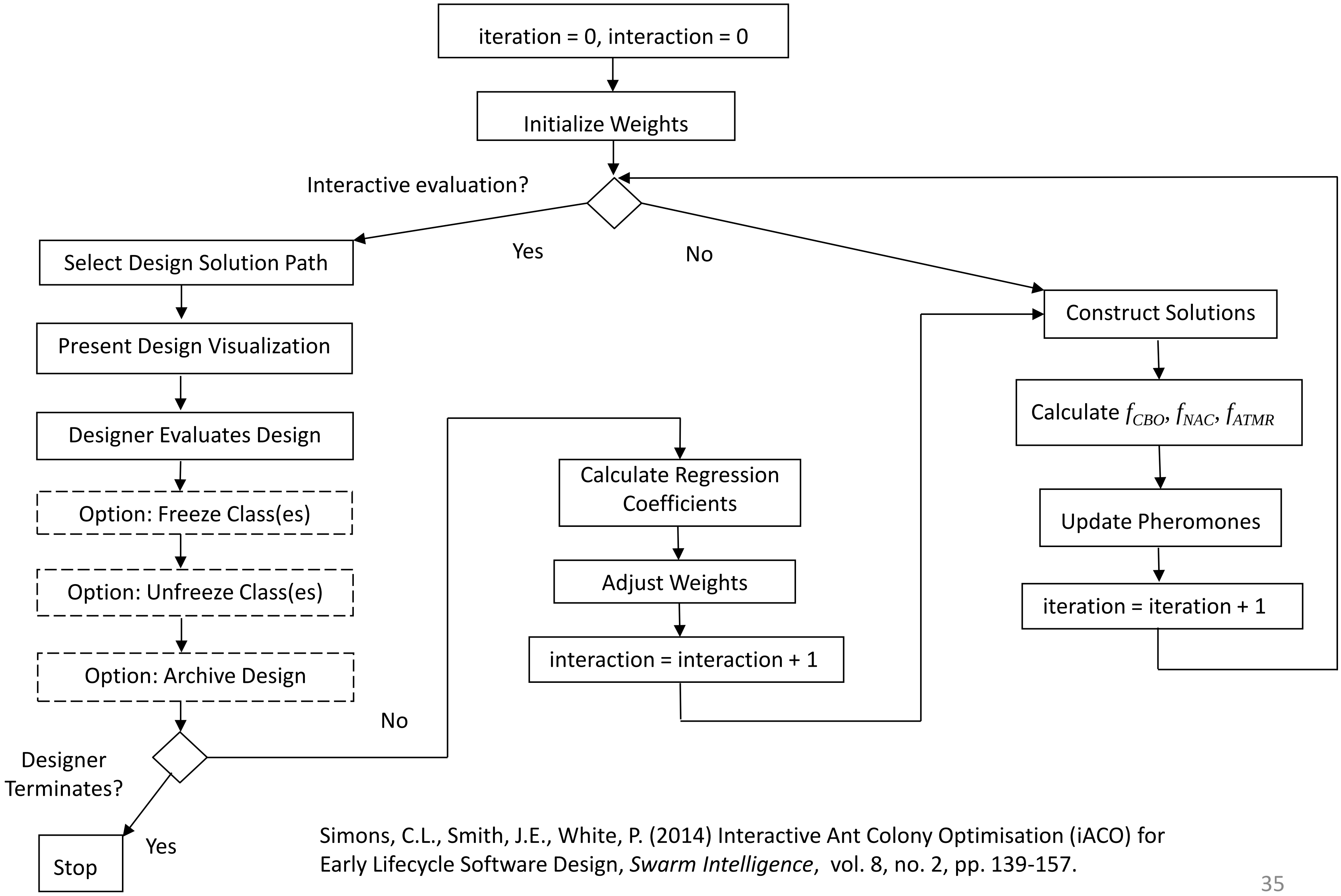
adjust weights

yes

stop

Simons, C.L., Parmee, I.C. (2012) Elegant Object-oriented Software Design via Interactive Evolutionary Computation, *IEEE Transactions on Systems*, *Man and Cybernetics – Part C*, vol. 42, no. 6, pp. 1797-1805.

Example (2) Interactive Ant Colony Optimisation



iteration = 0, interaction = 0
Initialize Weights
Interactive evaluation?
Yes
No
Select Design Solution Path
Present Design Visualization
Designer Evaluates Design
Option: Freeze Class(es)
Option: Unfreeze Class(es)
Option: Archive Design
Designer Terminates?
No
Yes
Stop
Calculate Regression Coefficients
Adjust Weights
interaction = interaction + 1
Construct Solutions
Calculate f_{CBO}, f_{NAC}, f_{ATMR}
Update Pheromones
iteration = iteration + 1

Simons, C.L., Smith, J.E., White, P. (2014) Interactive Ant Colony Optimisation (iACO) for Early Lifecycle Software Design, *Swarm Intelligence*, vol. 8, no. 2, pp. 139-157.

Challenges for Evolutionary Algorithms?

- Searching for strategies rather than instances
- Exploiting many-core computing
- Giving insight to software developers
- Optimising compilation and deployment
- ***Balancing Computation and Human Interaction***

Harman, M. (2012) "The Role of Artificial Intelligence in Software Engineering", *Proceedings of the First International Workshop on Realising Artificial Intelligence in Software Engineering (RAISE)* , pp. 1-6.

Some resources available

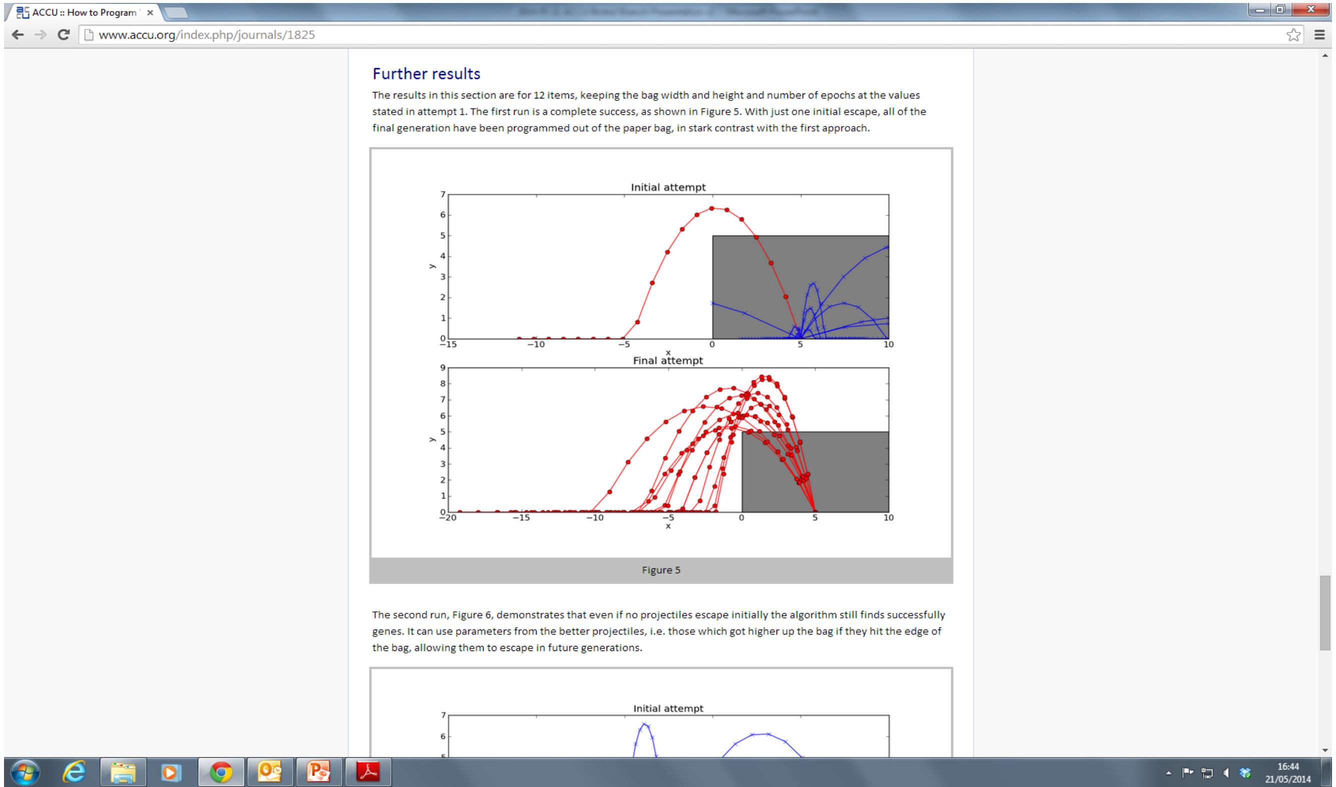
- Evolving Objects (EO): an Evolutionary Computation Framework (C++)
  - http://eodev.sourceforge.net/
- Open BEAGLE (C++)
  - https://code.google.com/p/beagle/
- ECF (Evolutionary Computational Framework) (C++)
  - http://gp.zemris.fer.hr/ecf/
- ECJ 21 (Java Evolutionary Computation)
  - http://cs.gmu.edu/~eclab/projects/ecj/
- JCLEC – Java Class Library for Evolutionary Computation
  - http://jclec.sourceforge.net/
- Etc. etc.

And finally…

there's even an article on a GA in Overload!

Buontempo, F. (2013) How to Program Your Way Out of a Paper Bag Using Genetic Algorithms. *Overload*, Iss. 118 (December 2013).

http://www.accu.org/index.php/journals/1825



Further results

The results in this section are for 12 items, keeping the bag width and height and number of epochs at the values stated in attempt 1. The first run is a complete success, as shown in Figure 5. With just one initial escape, all of the final generation have been programmed out of the paper bag, in stark contrast with the first approach.

Figure 5

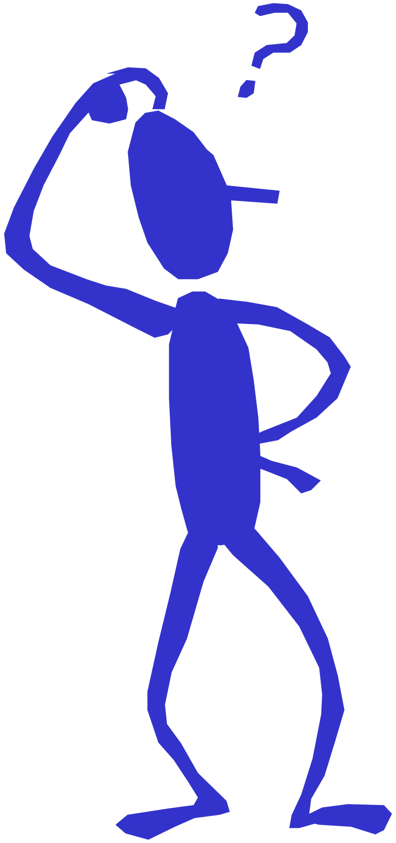
The second run, Figure 6, demonstrates that even if no projectiles escape initially the algorithm still finds successfully genes. It can use parameters from the better projectiles, i.e. those which got higher up the bag if they hit the edge of the bag, allowing them to escape in future generations.

Take away thoughts?

- Software Design
  - *complex, people-centred*
- Evaluation
  - *“multi-obsubjective” fitness evaluation*
- A fusion of software engineer and computer
  - *Partnership, ‘human-in-the-loop’*
  - *Combines human intuition with computational ‘intelligence’*

Thank you

- Any questions?



chris.simons@uwe.ac.uk

www.cems.uwe.ac.uk/~clsimons

@chrislsimons #ACCU2015