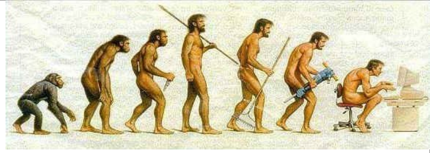
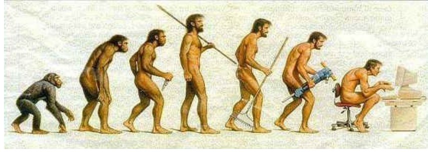


SERIOUS

How to Maintain Quality in Evolving Software Systems

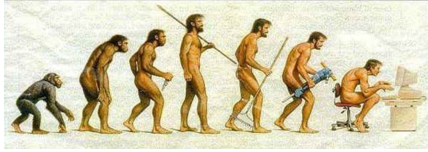
Juha Gustafsson

About Serious



- **Software Evolution, Refactoring, Improvement of Operational & Usable Systems**
  - **started 09/2005**
- **EU/ITEA project, TEKES funding in Finland**
- **Partners: Alcatel, U Antwerp, UP Madrid, Philips (Medical Systems, Applied Technologies), Surlog, Bertin Technology, Ibermatica**
- **Finnish partners: Nokia Research Center, U Helsinki, Calassa Labs, Tampere UT**
- ***http://www.hitech-projects.com/euprojects/serious/***

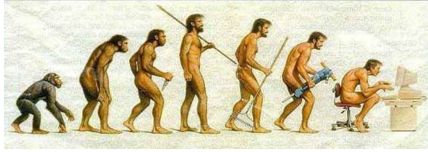
Research problem



- Software intensive systems evolve during their lifetime
- The size of software assets can be considerable
- During the creation phase, the overall quality of the product increases and reaches a defined level
- After that the software needs to be extended, for example, with new features
- Within that period, which will last until the product is phased out, the software quality will degrade if no special measures are taken to prevent this

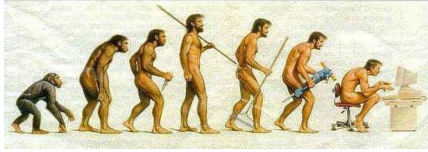
What can be done to prevent the deterioration of software quality?

Quality – what for?



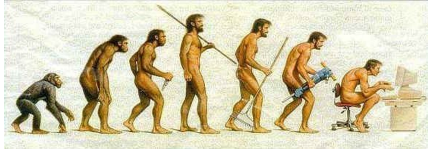
- Quality is a measure of the excellence of a product
- The higher the quality the better is the product
- One main objective of a product developer is to maximize the quality (at a certain cost)
- If we want to optimize the quality of a product, we have to be able to measure it first
- Different kinds of qualities:
    - Reusability of the code, Functionality, Reliability, Usability, Efficiency, Maintainability, Portability, ...
    - Can we optimize them all?

Serious goals



- Develop techniques, tools and models to improve quality aspects related to software evolution
- Develop processes and methods to improve the software development models in industry
- Define various quality aspects in a level of detail that it can be applied and/or used in the early phases of software development and continued to be useable in the evolution phase

Common problems



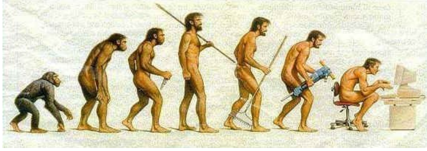
- Managing a large software repository

- Refactoring
  - Preparing for requirement changes and additions
  - Adding new features without affecting the existing software

- Keeping software product lines consistent

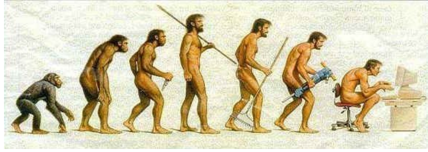
- Defining quality:
  - How can we measure the quality realistically?
  - Can we enhance all the quality features at the same time?
  - Are there relationships between *internal* and *external* qualities?
  - Are the tools used for measuring the quality reliable?

Example: Contradictory requirements?



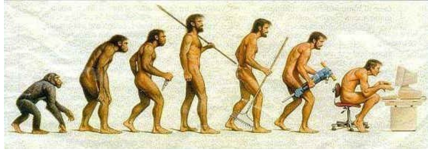
- While our main focus is on developing individual quality metrics, we should also study how they will work on the same context and resolve possible conflicts
- In many cases we have to be content with less than optimal solution in respect to individual quality aspects. This is true especially with product families that frequently need to be prepared for a wide variety of requirements that may change over time
- When refactoring existing software to accommodate new or altered requirements we should take into account the effects these modifications have on other quality aspects

Contradictory requirements? (cont'd)



- To what extent is it possible to accommodate conflicting requirements?
- How can we evaluate how good a particular design solution is relative to our quality requirements?
- How does the software process handle contradictory requirements?

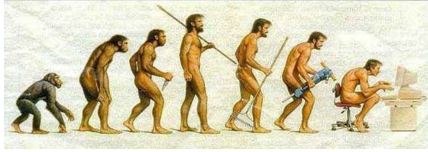
Coupling



"Coupling quantifies dependencies between software entities"

- **Evaluation of coupling metrics**
  - dependency models
  - system coupling
- **Developing and evaluating new coupling techniques**
  - quality vs. quantity
- **Applicability of existing coupling metrics at architecture level**
  - case studies

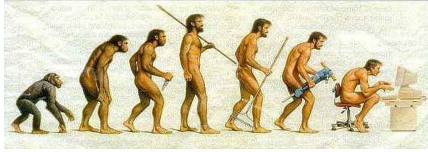
Variability



“Variability is the ability of a software system to be changed or customized”

- **Defining variability metrics**
  - variation types, granularity of SW artifacts
- **Variability management in SW process**
  - practices: audits, feature models
  - artifacts: variability models
- **Aspect-Oriented approaches**
  - modeling features as aspects
  - common mechanisms in product lines
- **Cost of variability?**

Future work



- Design and implementation of coupling and variability metrics
- Case studies (UH/NRC)
  - started 01/2007
- Collaboration with other Serious tasks
  - quality models
  - refactoring?