

The background of the slide features a close-up, slightly blurred image of a clock face and its mechanical components, such as gears and a winding knob. The clock face shows numbers 12, 1, 2, and 3, with the hands pointing towards the 10 and 2 o'clock positions. The overall color scheme is a warm, reddish-brown hue.

# How to Estimate Software Size and Effort in Iterative Development

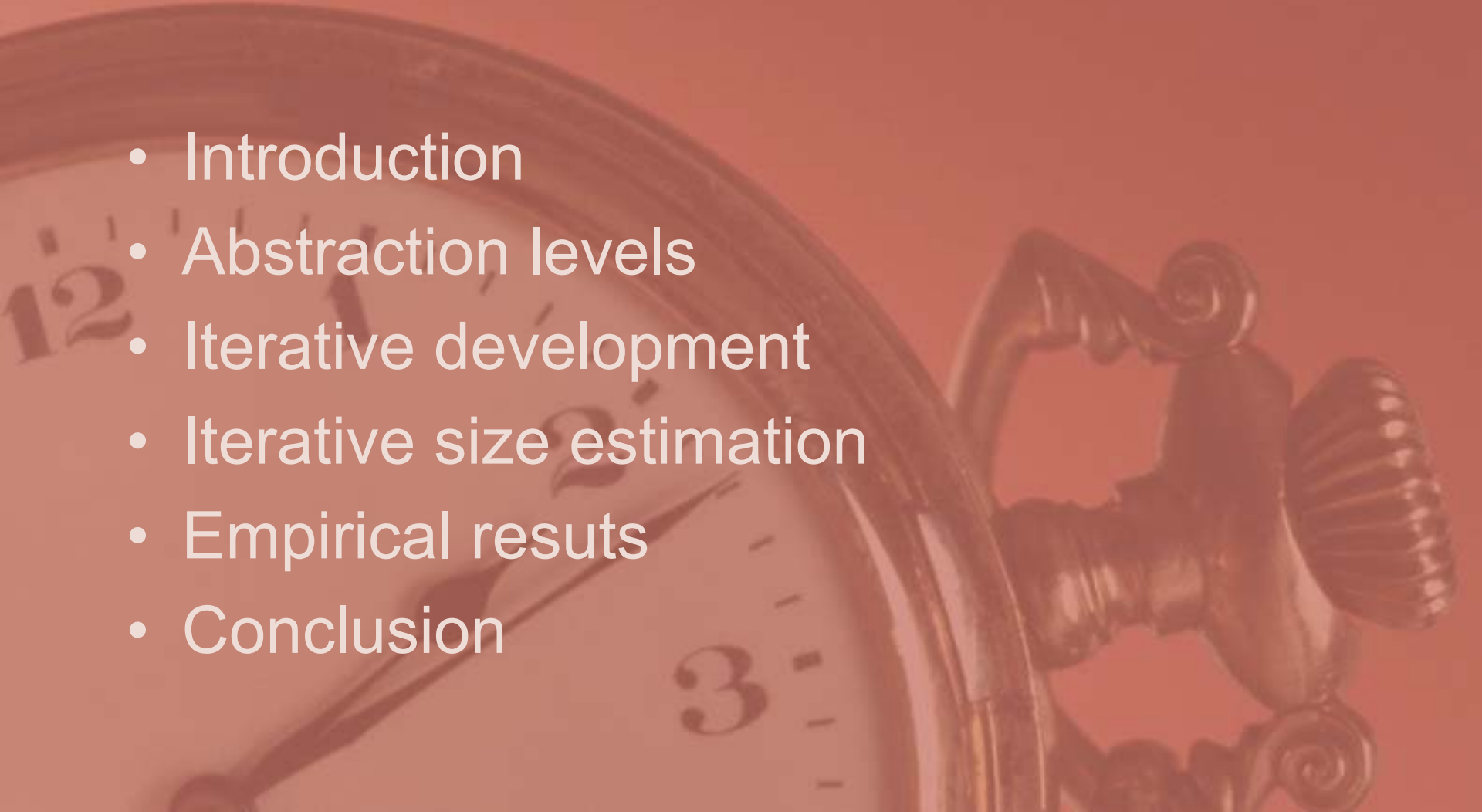
Aleš Živkovič

UM FERi Maribor

Smetanova ulica 17, SI-2000 Maribor, Slovenia, EU

[ales.zivkovic@uni-mb.si](mailto:ales.zivkovic@uni-mb.si)

# Content

- Introduction
  - Abstraction levels
  - Iterative development
  - Iterative size estimation
  - Empirical results
  - Conclusion
- 

# Size Estimation Methods

- Two major groups:
  - analogy based
  - model (algorithmic) based



# Analogy based





# Model based



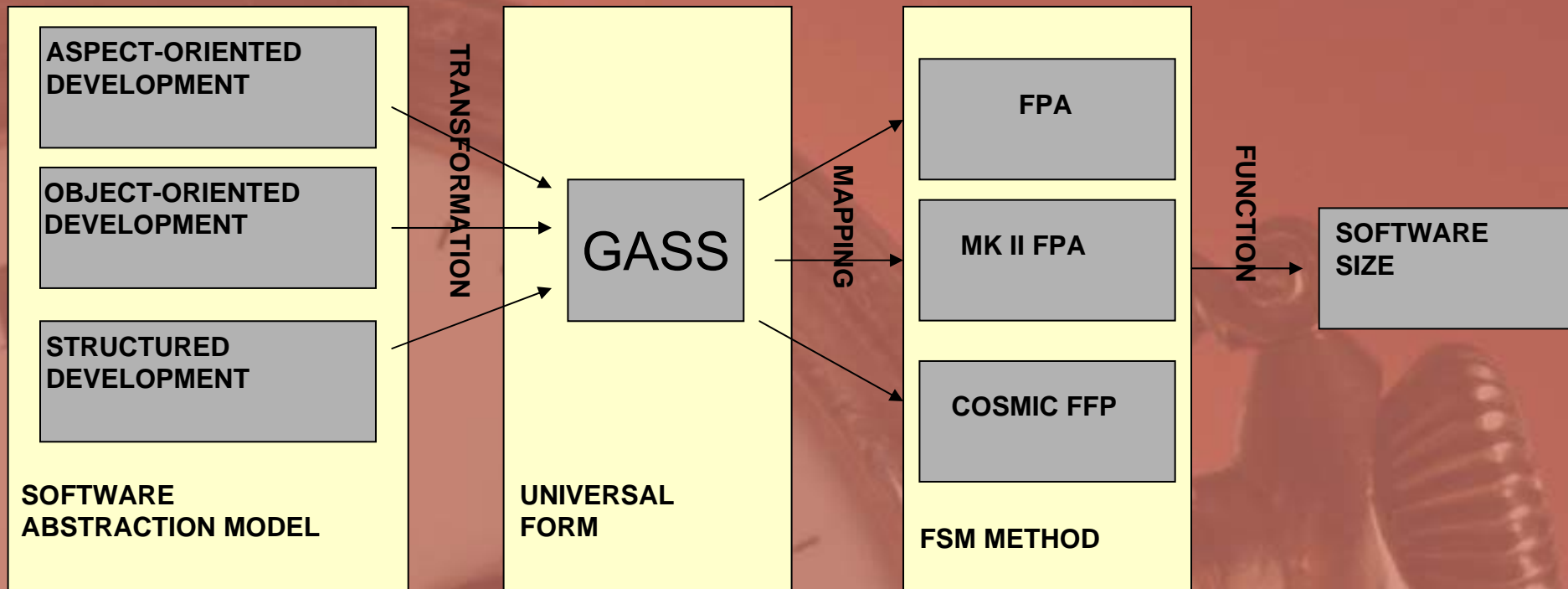
$$\text{Effort} = \text{surface} * 1,1$$



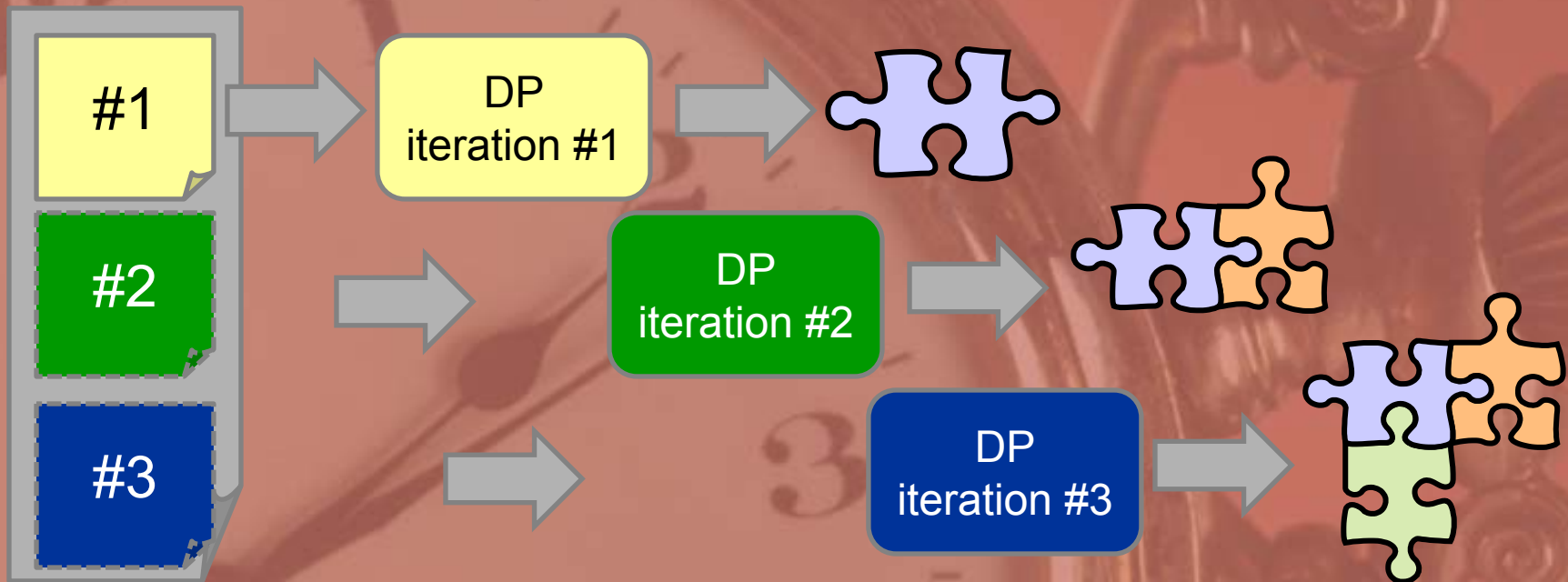
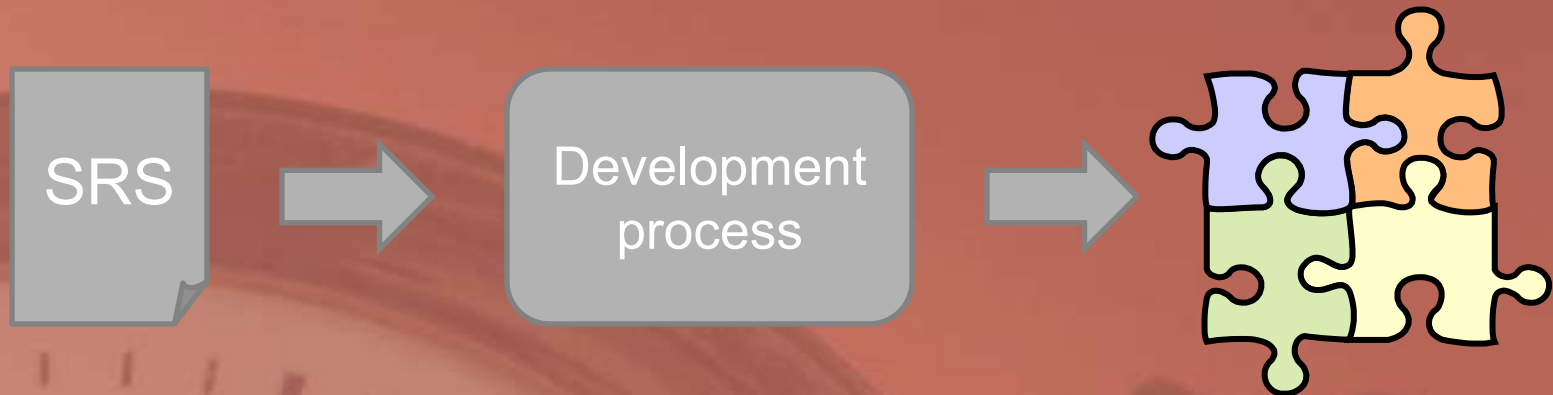
$$\text{Effort} = \text{surface} * 1,1 + (\text{surface}/5) * 2,5$$



# Universal Size Estimation Process



# The iterative problem

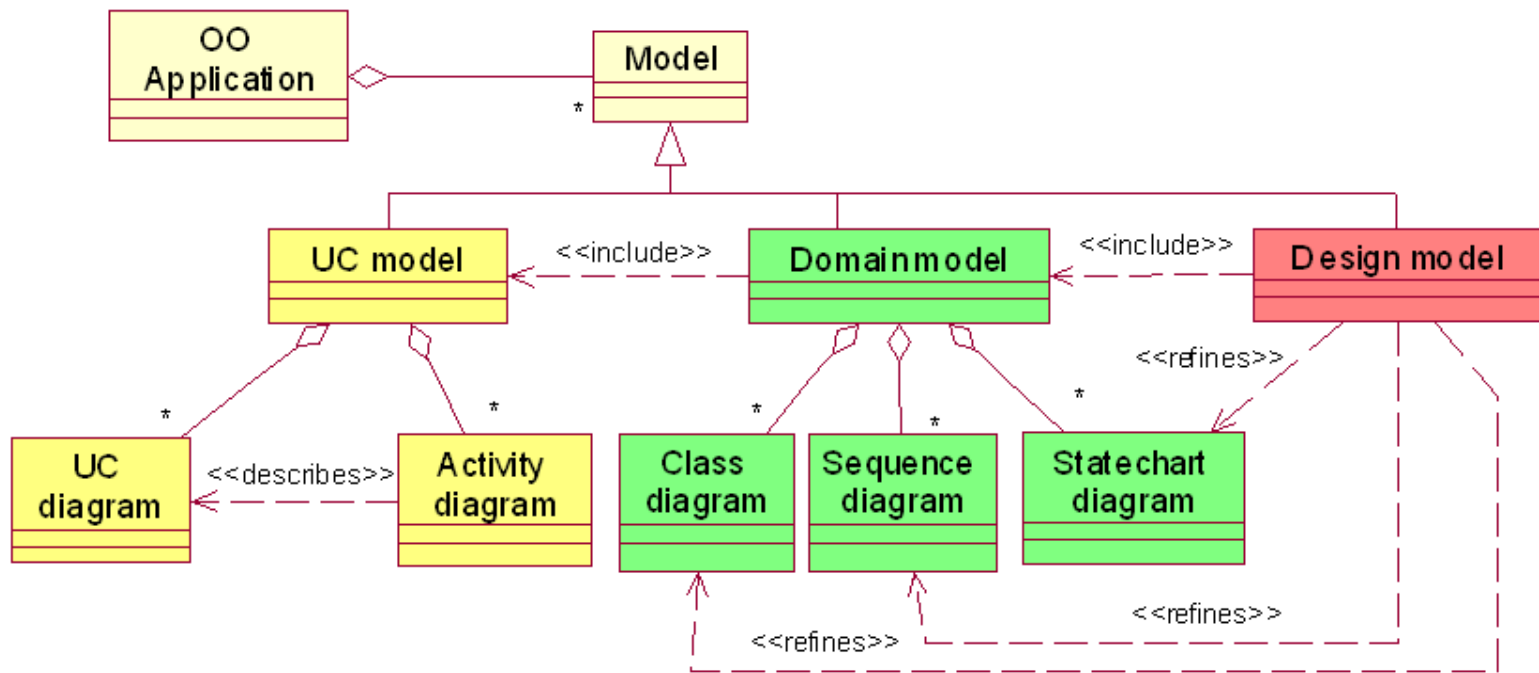


# The solution for iterative size estimation

- Representation of software system at different abstraction levels
- Combining early size estimates with more accurate size estimates through the iteration
- Improving the estimate with lessons learned in previous iterations.



# Abstraction levels with UML



# The idea - simplified view

EARLY ESTIMATION

ITERATION #1

ITERATION #2

I #1

UC1

UC2

CLASS  
DIAGRAM  
(UC1 & UC2)

EFFORT

I #2

UC3

UC3

CLASS  
DIAGRAM  
(UC3)

$\text{Size} = \text{Size}(\text{CD}) + \text{Size}(\text{UC3}) + d$

$\text{Size} = \text{Size}(\text{UC1} \sim \text{UC3})$

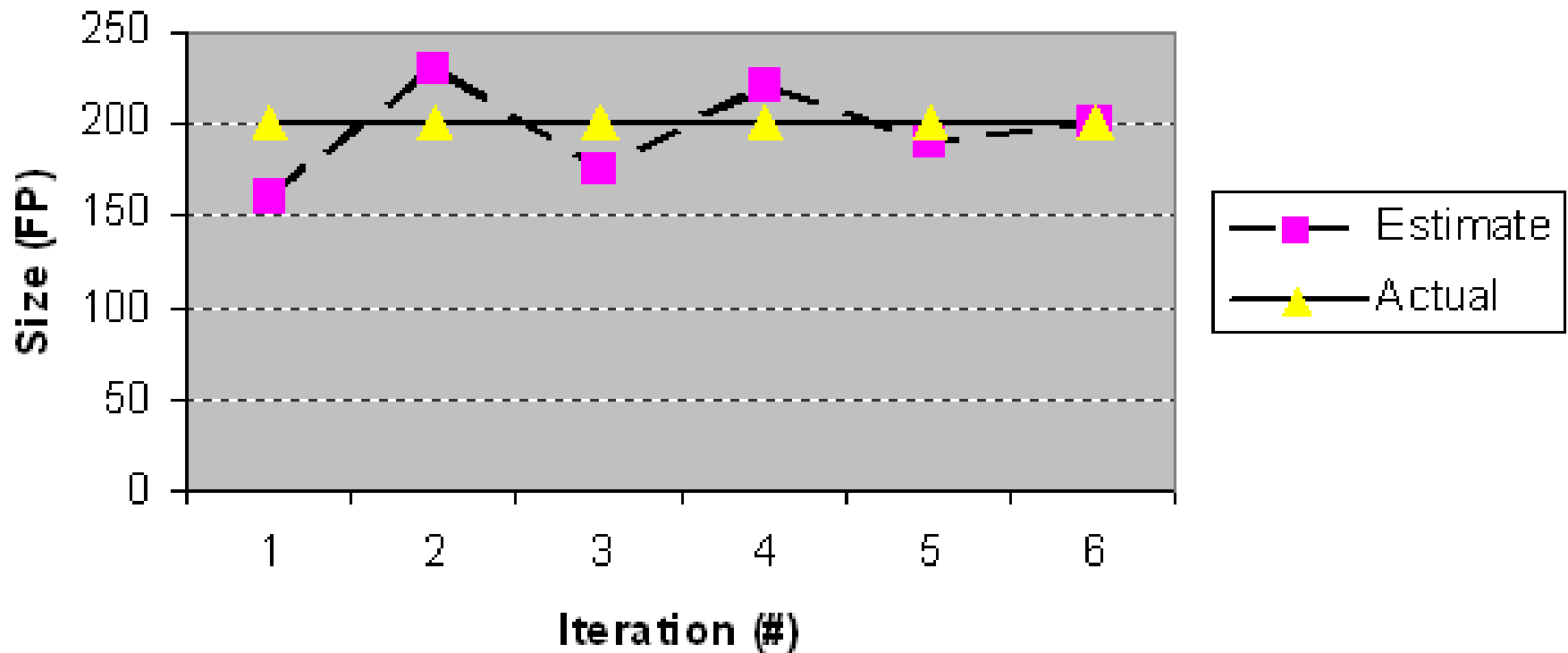
$\text{Size} = \text{Size}(\text{Effort}) + \text{Size}(\text{CD})$

# The problem of true value

- What is the correct software size?
- It is easy to measure distance! What about software size?
- Usual approach
  - Effort = Size \* Productivity
  - Use of industry repositories- **ISBSG\***

# Accuracy improvement

## Improvement of Estimates through Iterations



# Case study

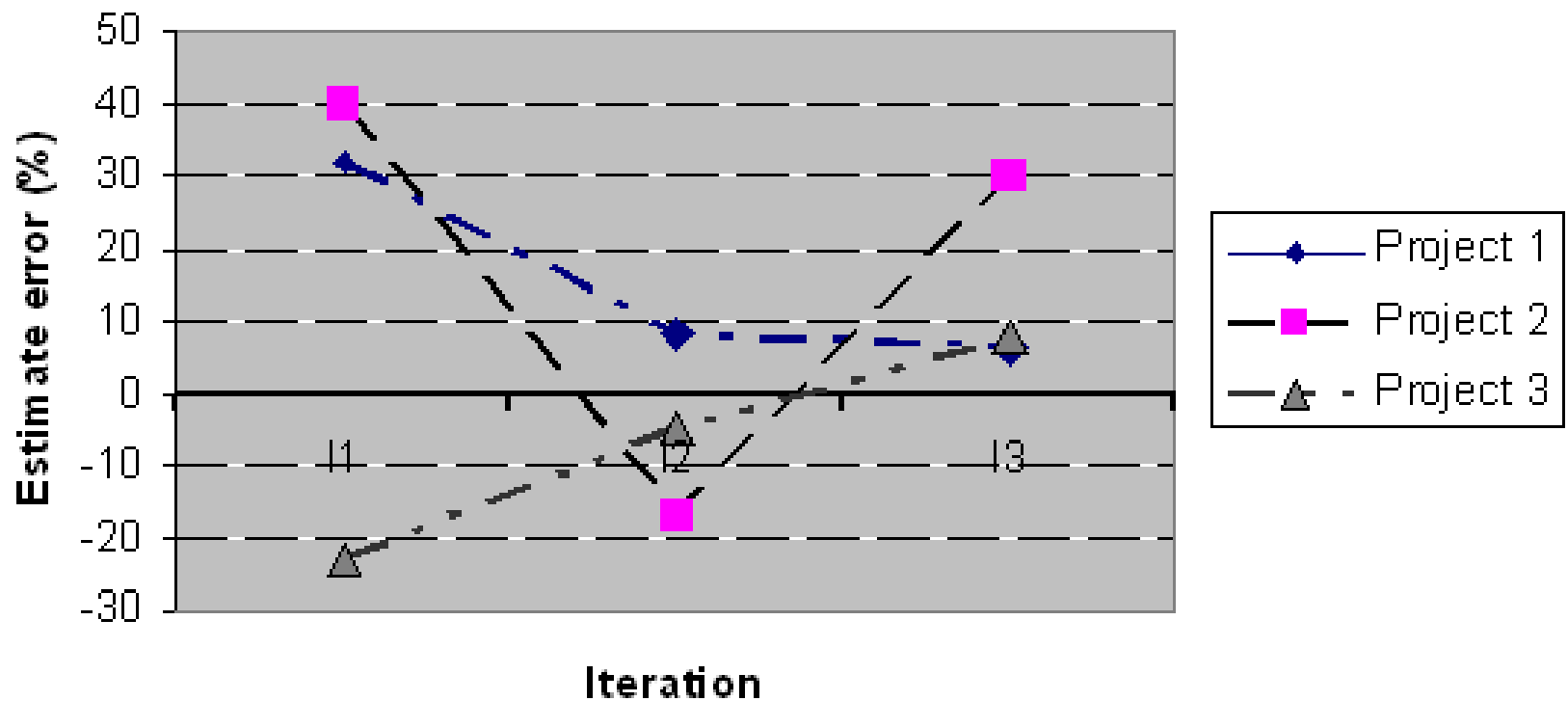
- Approach tested on 3 projects
- Programming language: Java
- Projects' statistics:

	<b>SLOC</b>	<b>NOC</b>	<b>NOM</b>	<b>GUI</b>	<b>Effort (h)</b>
<b>Project 1:</b>	1946	13	37	16 JSP	166
<b>Project 2:</b>	791	5	57	19 JSP	166
<b>Project 3:</b>	1307	20	30	9 SWT	205



# Estimation Error

## Change in Estimation Error through Iterations



# Discussion of results

- Although the sample is small, the results are promising
- The change in productivity could influence the accuracy - the problem of true value!
- The method could help project managers manage and report on time before it is too late (consistent with the PRINCE 2 manage by exception approach)

# Conclusion

- Size estimation in an iterative environment is difficult
- Traditionally well known and accepted methods could not be used
- The presented approach is a combination of two popular methods
- For the application of the method on industry projects, tool support is crucial.
- Open to help vendors implement the approach in development tools that support UML 2.0