review

what?

“ **Process** or **Meeting**

during which a **software product** is examined by

a project personnel ,
users ,
user representatives ,

managers ,
customers ,
or other interested parties

for **comment** or **approval** ”

http://en.wikipedia.org/wiki/Software_review

The earlier errors are found

the lower costs

correcting errors
precisely

http://www.slideshare.net/oanafeidi/reviews-checklists

Review Target

review target

1. Have more **understandable** project

2. **Saving** implementation time

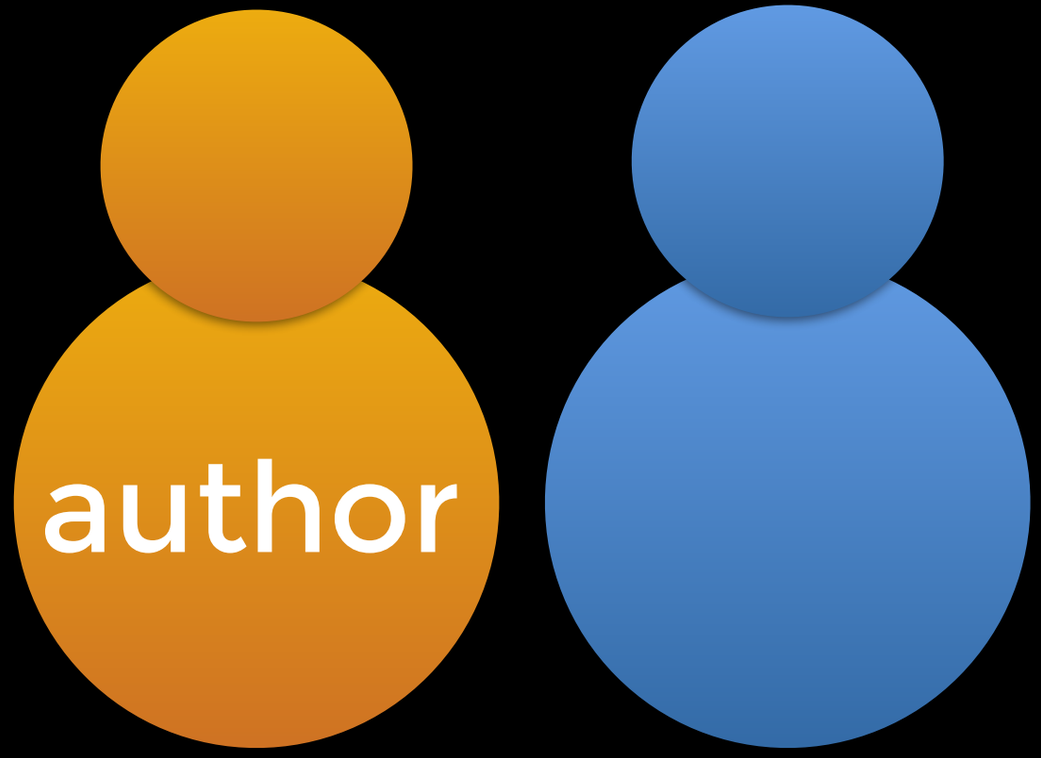
3. **Improving** the efficiency of the reviews

http://www.slideshare.net/oanafeidi/reviews-checklists

Type of Review

type of review

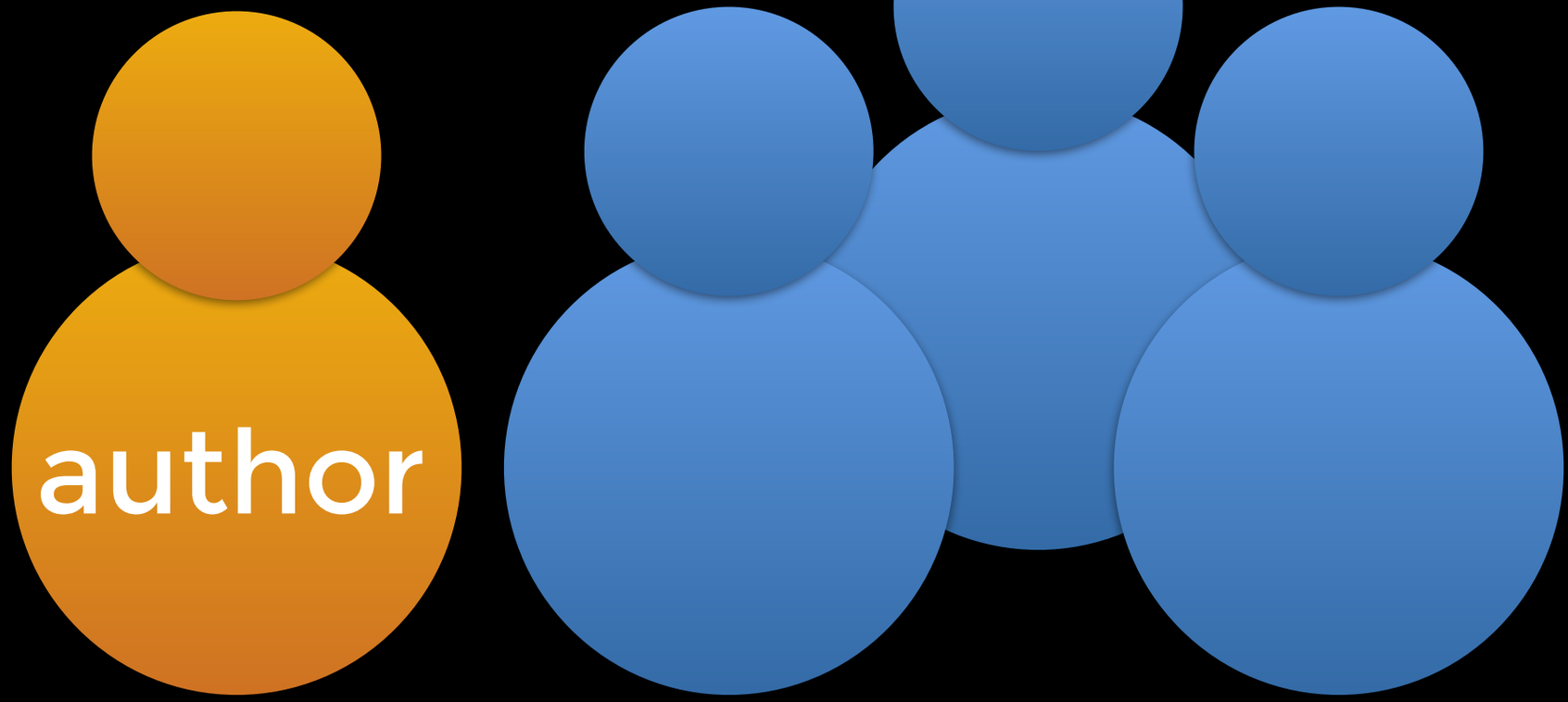
1. Informal Review



http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

type of review

1. Informal Review



Peer review : Reviewed by people who are at the same level

http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

type of review

2. Inspection

Very Formal Type

author

http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

type of review

2. Inspection

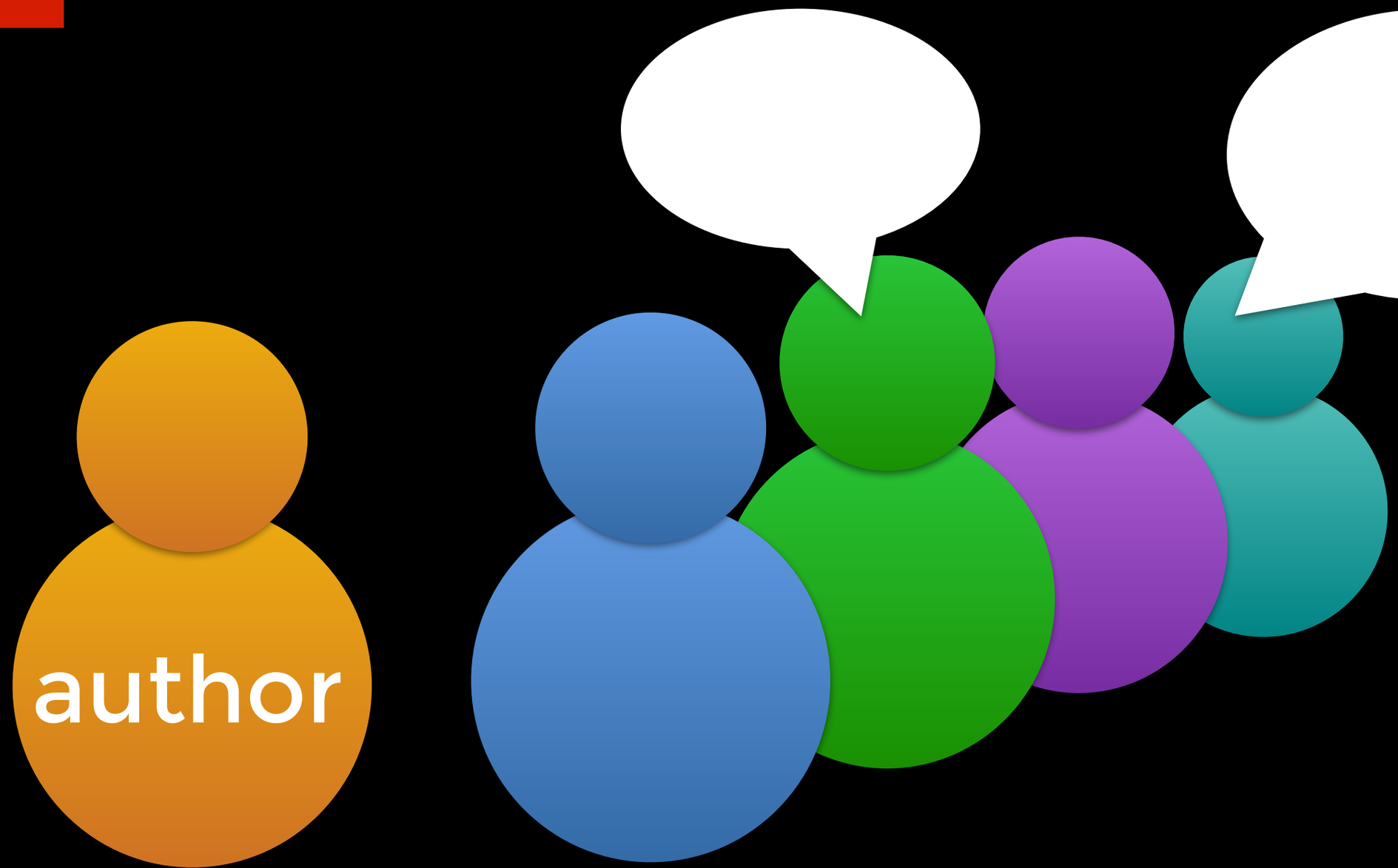
Identify anomalies

How to improve?

http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

type of review

3. Walkthrough



http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

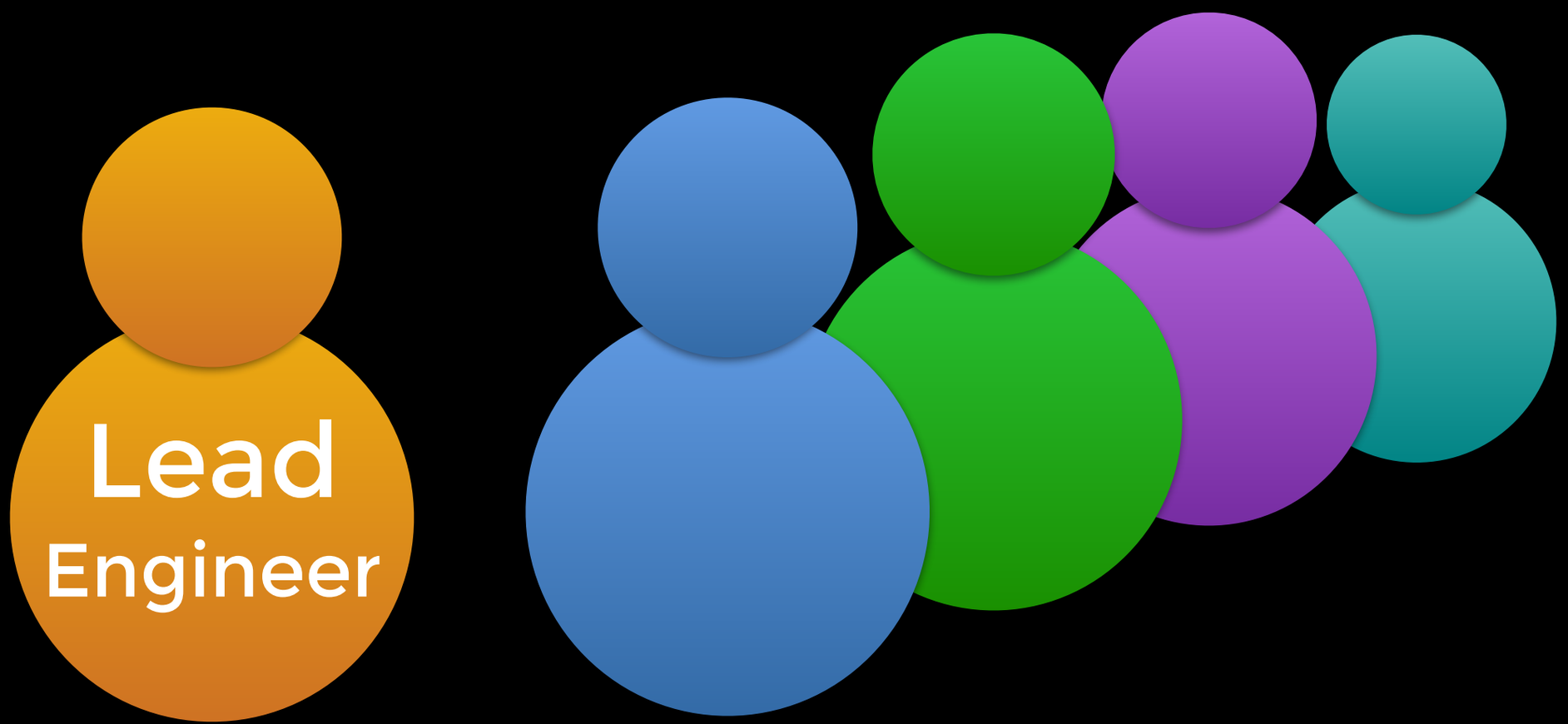
4. Technical Review

type of review

http://74.220.219.56/~nursetop/wp-content/uploads/2011/03/levels-of-evidence1.jpg

type of review

4. Technical Review



http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

4. Technical Review

type of review

Team with evidence to confirm

Suitability

Stick to regulations

Change concerned

http://profs.etsmtl.ca/claporte/English/Enseignement/CMU_SQA/Notes/Revues/IEEE_Std_1028_Reviews.pdf

Advantage

advantage

1. Improve schedule predictability

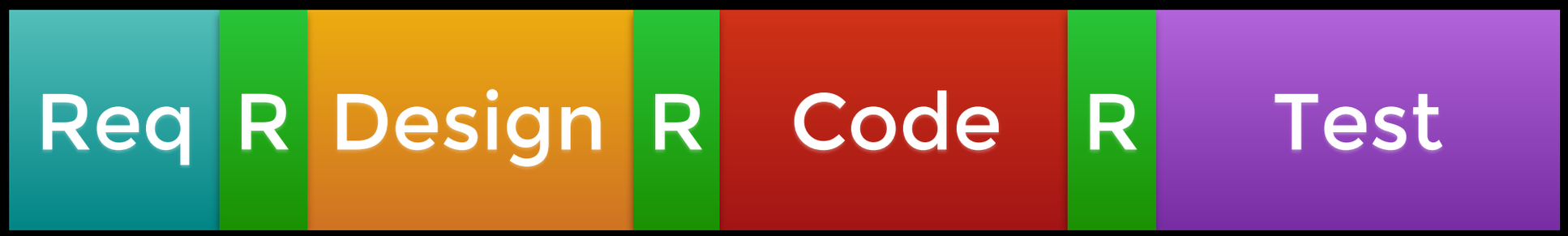
no reviews

http://www.slideshare.net/oanafeidi/reviews-checklists-11467857#btnPrevious

advantage

1. Improve schedule predictability

reviews



http://www.slideshare.net/oanafeidi/reviews-checklists-11467857#btnPrevious

advantage

2. Reduce rework

http://www.slideshare.net/oanafeidi/reviews-checklists-11467857#btnPrevious

advantage

3. Proactive Tests

http://www.slideshare.net/oanafeidi/reviews-checklists-11467857#btnPrevious

advantage

4. Training

http://www.slideshare.net/oanafeidi/reviews-checklists-11467857#btnPrevious

Formal Reviews

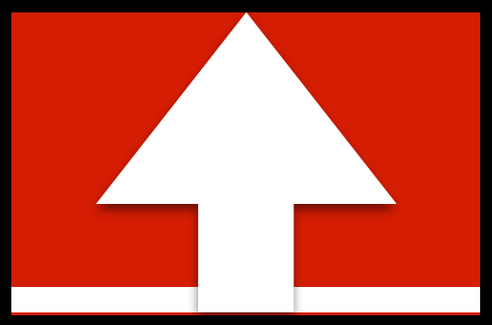
standard

IEEE

Institute of Electrical and Electronics Engineers

Std. 1028

IEEE Standard for Software Reviews and Audits



Based on IBM's **Software Inspection** process

IEEE Std 1028™-2008
(Revision of
IEEE Std 1028-1997)

IEEE Standard for Software Reviews and Audits

Sponsor

Software & Systems Engineering Standards Committee
of the
IEEE Computer Society

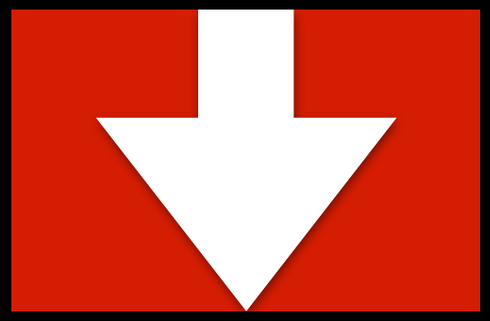
Approved 16 June 2008

IEEE-SA Standards Board

Authorized licensed use limited to: Kasetsart University provided by UniNet. Downloaded on February 03,2014 at 05:48:14 UTC from IEEE Xplore. Restrictions apply.

IEEE Std. available to download at
http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4601582

references[1]



Download IEEE Std. 1028
(from ieee.org[1])

Read more about Fagan's
Inspection (from wikipedia.org[2])

[1]: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4601582
[2]: http://en.wikipedia.org/wiki/Michael_Fagan_(software_designer)

Guideline

IEEE Std. 1028

Types of Review

Management Reviews

Technical Reviews

Inspection

Walk-Throughs

Audits

d. 1028

processes of the standard (1)

0. Entry Evaluation

1. Management Preparation

2. Planning the Review

processes of the standard (2)

3. Overview of Procedure

4. Preparation

Individual

5. Examination

Group

processes of the standard (3)

6. Rework/Follow-Up

7. Exit Evaluation

Source: http://en.wikipedia.org/wiki/Software_review

management review

Leadership Lead by **Manager**

Objective Evaluation of **Software Process** (eg. Development Process)

Output Management report

technical review

Leadership Lead by **Lead Engineering**

Objective Evaluation of **Software Product** (eg. Development Process)

Output Technical report

inspection

Leadership Lead by **Trained Facilitator**

Objective Examination **defects** and identify **anomalies**

Output Defect list

walk-through

Leadership

Lead by **Facilitator** or
Author

Objective

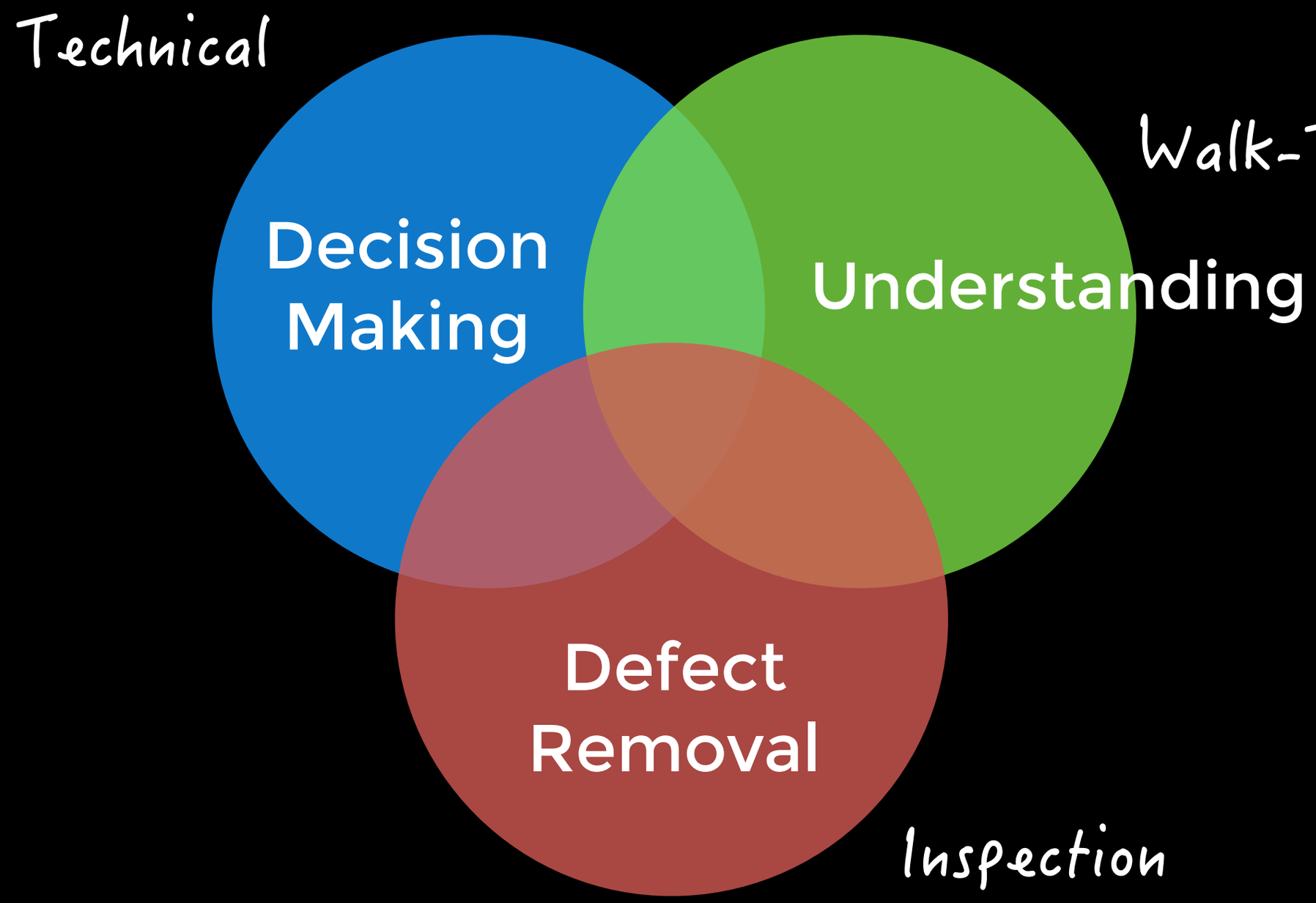
Static analysis technique
of a software product

Output

Report

Source:
http://en.wikipedia.org/wiki/Software_review

focus of types of review



Gilb & Graham, Inspection Course notes, September 1995

review differences

	Management Review	Technical Review	Inspection	Walk-Through
Leadership	Manager	Lead Eng.	Trained Facilitator	Facilitator or Author
Objective	Ensure Progress	Ensure Progress	Ensure Progress	Ensure Progress
No. of Members	Unlimited	Unlimited	Unlimited	Unlimited
Output	Management Report	Management Report	Management Report	Management Report

Source: http://profs.etsmtl.ca/

TASK

5 minutes

```
import java.io.BufferedReader;
import java.io.IOException;

public class Main {
	/**
	 * @param args
	 * @throws IOException
	 */
	public static void main(String[] args) throws IOException {
		String filename = args[0];
		BufferedReader reader = readFile(filename);
		double xavg = calculateData(reader);
		System.out.println("Average" + xavg);
	}

	public static BufferedReader readFile(String filename) {
		BufferedReader reader = new BufferedReader(new FileReader(filename));
		return reader;
	}

	public static double calculateData(BufferedReader reader) throws IOException {
		double SUmX = 0;
		int count = 0;
		String line = reader.readLine();
		while(line == null) {
			count++;
			double temp = Double.parseDouble(line);
			sumx += temp;
		}
		return sumx/count;
	}
}
```

Includes	Verify that the includes are complete.	x
Initialization	Check variable and parameter initialization. - at program initiation - at start of every loop - at class/function/procedure entry	✓
Calls	Check function call formats. - pointers - parameters - use of '&'	✓
Names	Check name spelling and use. - Is it consistent? - Is it within the declared scope? - Do all structures and classes use '.' reference?	x
Output Format	Check the output format. - Line stepping is proper. - Spacing is proper.	x
() Pairs	Ensure that () are proper and matched.	✓
Logic Operators	- Verify the proper use of ==, =, ll, and so on. - Check every logic function for ().	x
Line-by-line check	Check every line of code for - instruction syntax - proper punctuation	✓

CASE STUDY: Meeting Place

(Cisco's computer-based audio and video teleconferencing software)

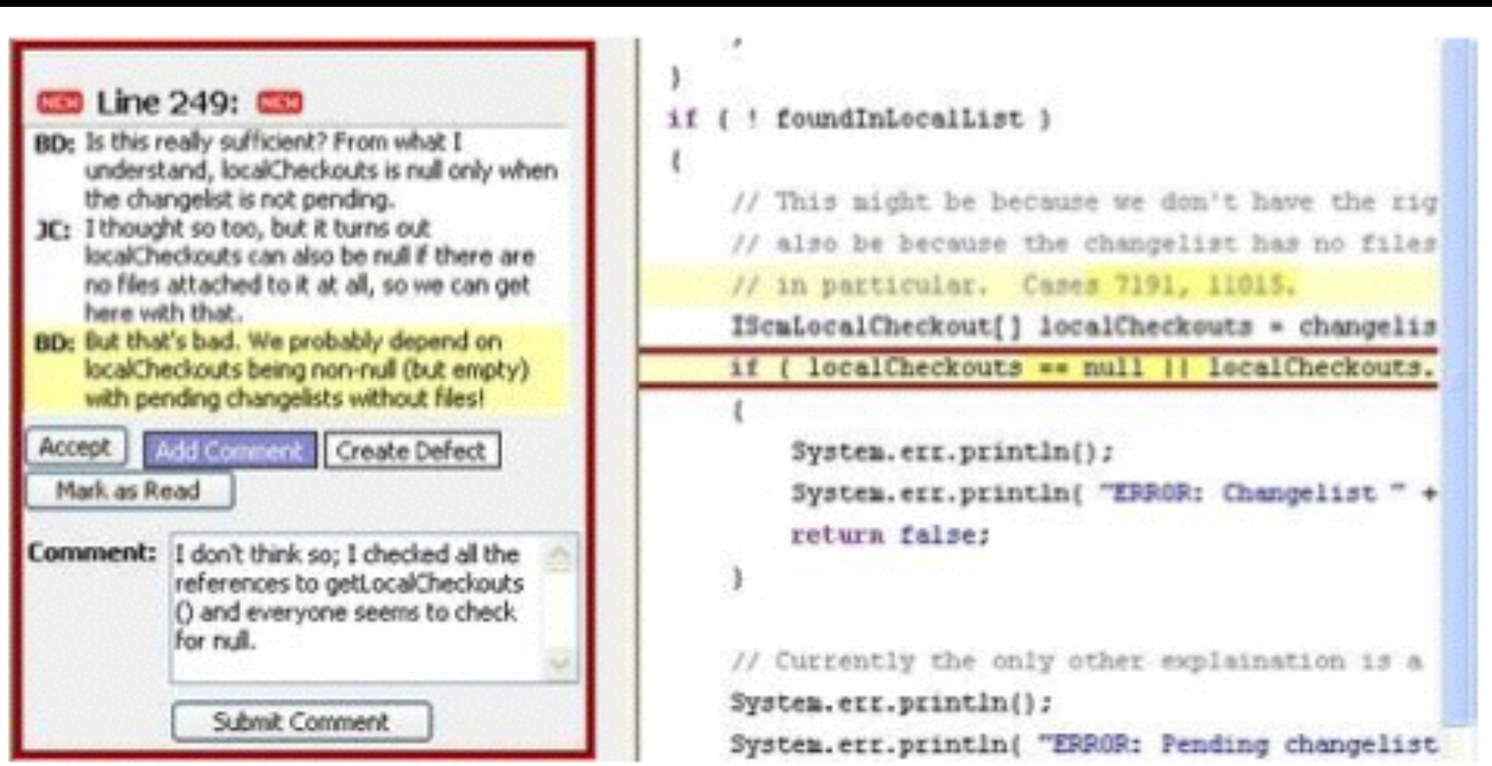
2500 Reviews.
50 Developers.
3.2M Lines of Code.

REAL SOFTWARE

How reviews were conducted?

CodeCollaborator

- Defects were logged by comment
- Collect process metrics automatically (LOC, number of defects, amount of person-hours spent in the review)

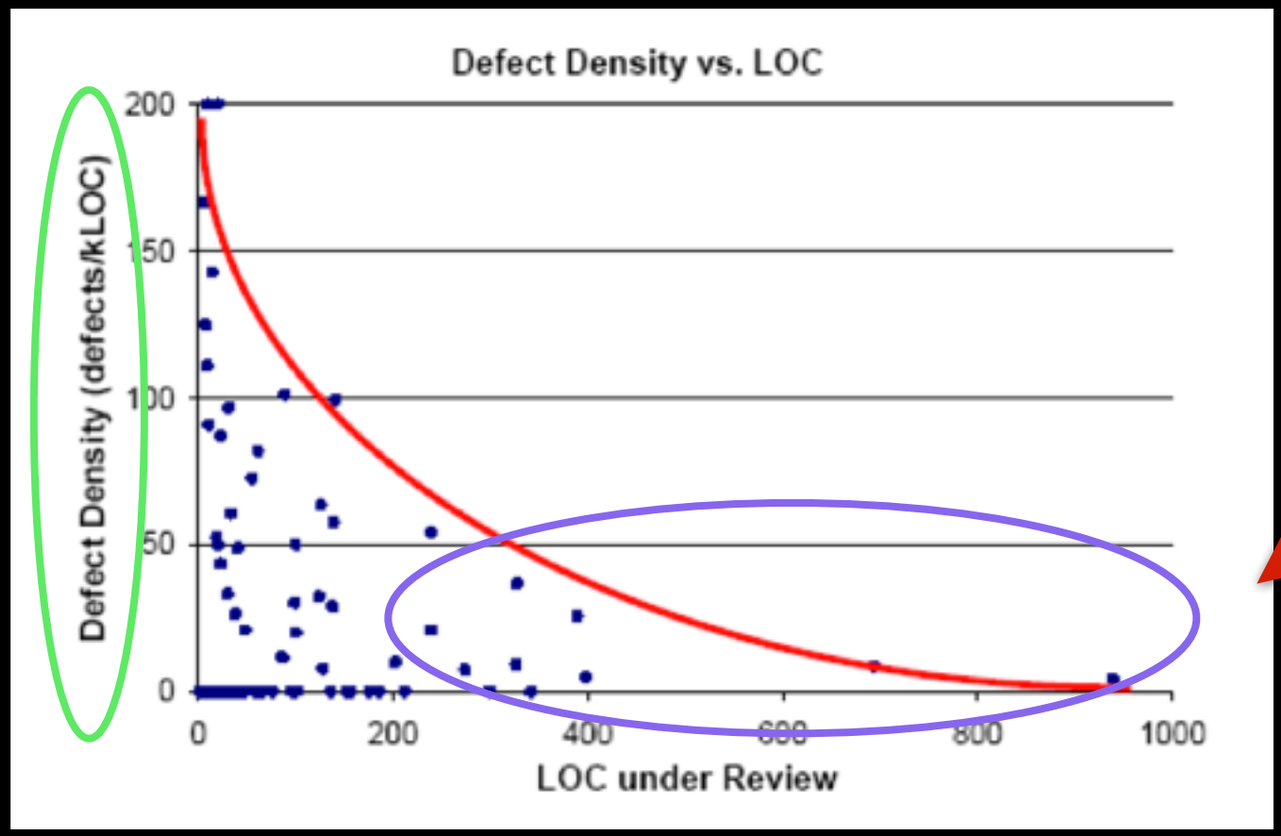


CONCLUSION

Don’t review too much code at once (< 200 - 400 LOC)

#1

review effectiveness

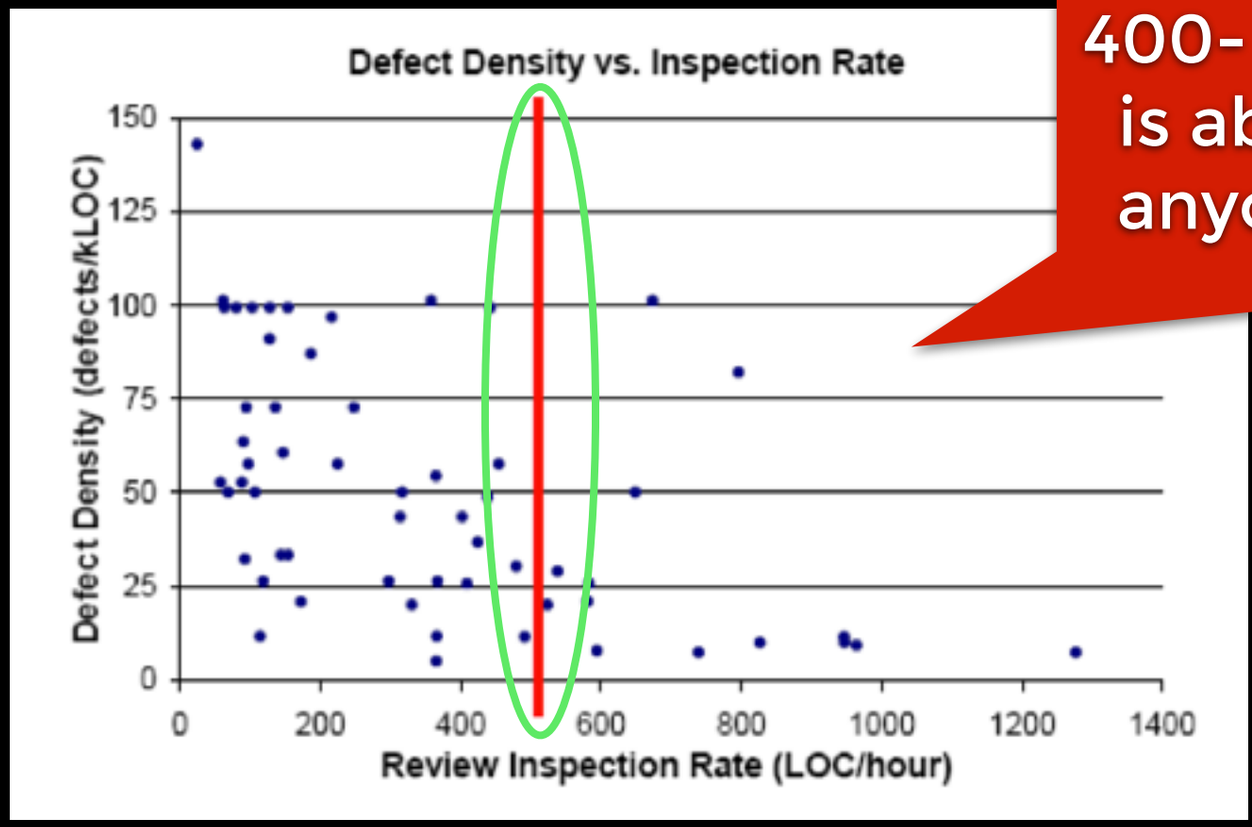


Defect density decreased when LOC under inspection went above 200

This data shows exactly where the boundary is between "OK" and “too much." 200 LOC is a good limit; 400 is the absolute maximum.

Take your time (< 500 LOC/hour)

#2



400- 500 LOC/hour is about as fast as anyone should go

the general result is not surprising: If you don't spend enough time on the review, you won't find many defects.

Spend less than 60 minutes reviewing

#3

< 1 hour

In fact, it's generally known that when people engage in any activity requiring concentrated effort, performance starts dropping off after 60-90 minutes.

Summary

1. Lightweight-style reviews are effective and efficient.
2. Review fewer than 200-400 LOC at a time
3. Aim for an inspection rate of less than 300-500 LOC/hour
4. Take enough time for a proper, slow review, but not more than 60-90 minutes

source: http://smartbear.com/CMSPages/GetFile.aspx?guid=d9f14ad2-1943-418f-a714-7aff514dec32

Discussion

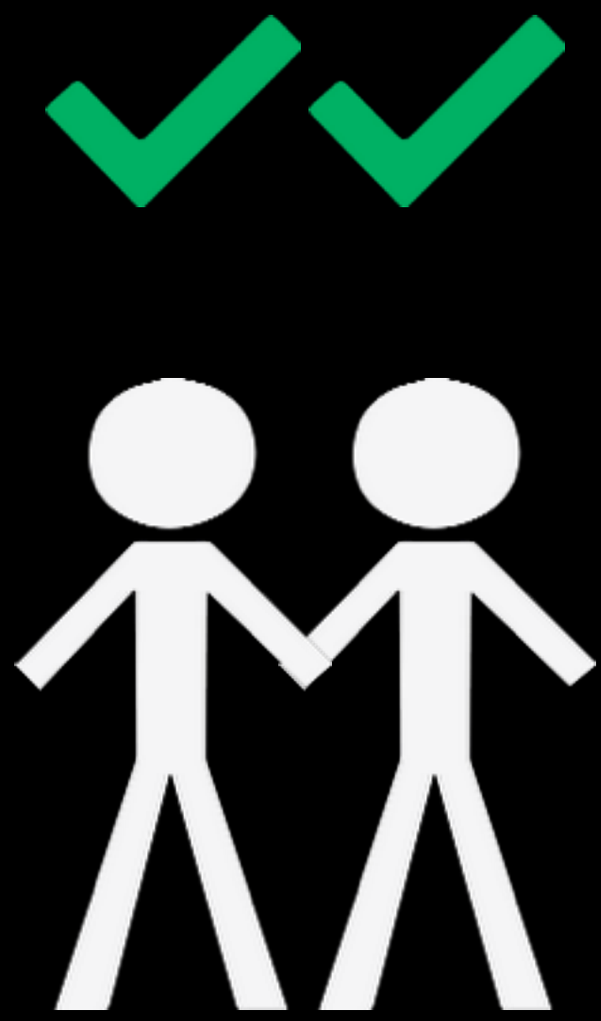
Review VS Testing

“Testing is essential”

“This maybe the same with review”

WHY?

Pair Programming



Single or Double?