

The background of the slide features a large, faint watermark of the University of Bologna seal. The seal is circular and contains the text 'UNIVERSITAS STUDIORUM BOLOGNENSIS' around the perimeter. In the center, there is a shield with a cross and the word 'LIBERTAS' above it, flanked by two figures.

A Double Comparative Study: Process Models and Student Skills

Paolo Ciancarini
(with C. Dos and S. Zuppiroli)
Department of Computer Science
University of Bologna – Italy
26th CSEE&T - S.Francisco, USA
May 2013



Summary of the talk

- Structuring a sw engineering lab course for a study on processes and skills
- Comparing the processes
- Comparing the skills
- Evaluation
- Related works
- Conclusions



Our goal

- Study how students perform with different process models
- Comparisons by process model and by student skills (classified by background)
- A few related works tell about similar experiments but with different settings



Related works

- Alfonso and Botia. An Iterative and Agile Process Model for Teaching Software Engineering. In Proc. 18° CSEET, Ottawa, Canada, 2005.
- Benediktsson, Dalcher, and Thorbergsson. Comparison of Software Development Life Cycles: A MultiProject Experiment. *IEE Proceedings - Software*, 153(3):87–101, June 2006.
- Hashmi and Baik. Software Quality Assurance in XP and Spiral - A Comparative Study. In Proc. IEEE Int. Conf. on Computational Science and its Applications (ICSSA), 2007.
- Ji and Sedano. Comparing Extreme Programming and Waterfall project results. In Proc. 24th CSEET, Waikiki, Hawaii, 2011.
- Layman, Williams, and Cunningham. Exploring Extreme Programming in Context: An Industrial Case Study. In Proc. Conf. Agile Development, 2004



Our SE class

- Third year undergraduate students
- Formal effort: 9 credits (225 hours)
- Duration: six months (oct..dec + mar..may)
- In 2011-12 about 100 students, from two different degrees:
 - Computer Science (**CS**)
 - Informatics for Management (**IM**)



Two different degrees

Computer Science



■ INF
■ Mat
■ Econ&Man
■ Other

Informatics for management



■ INF
■ Mat
■ Econ&Man
■ Other

	INF	Mat	Econ&Man	Other	Total
Computer Science	120	36	0	24	180
Informatics for management	81	26	50	23	180

A. Bolognesi, P. Ciancarini, and R. Moretti. On The Education of Future Software Engineers. In P. Inverardi and M. Jazayeri, editors, Software Engineering Education in the Modern Age: Challenges and Possibilities, PostProceedings of ICSE '05 Education and Training Track, volume 4309 of Lecture Notes in Computer Science , pages 186–205. Springer-Verlag, Berlin, 2006.



The SE course

- 27 lectures, 10 lab sessions
- Abstract of course syllabus:
 - Processes (especially RUP) 10h
 - Requirements engineering 8h
 - OO design with UML (a la Larman) 20h
 - Design patterns 10h
 - Measuring software quality 6h
- Exam: written test + project presentation



The project lab

- Four students per team (preferably)
- Each team got assigned a product and a process
- Four different products to develop
- Three different process models: **Waterfall**, Iterative (**RUP**), Agile (**XP**); the students were told this was a study to compare different process models
- Each team had role of “the client” for another team
- Each process included assignments grouped (and graded) by development phase



The products

Product	Average of total LOC
Carpooling	3664
Cinema	2842
Hospital system	4133
Medical system	6346



Comparing the processes

We focussed on the following process measures:

- duration and effort;
- size and quality of documentation;
- internal and external software qualities.



Comparing the processes

Duration and effort

Main result: XP more productive than Waterfall and Spiral

	Waterfall	Spiral	XP
Final delivery delay (days)	1.6	-1.2	0.4
Average total effort (hours)	457	367	330
Average total LOC	4469	2671	5125



Comparing the processes

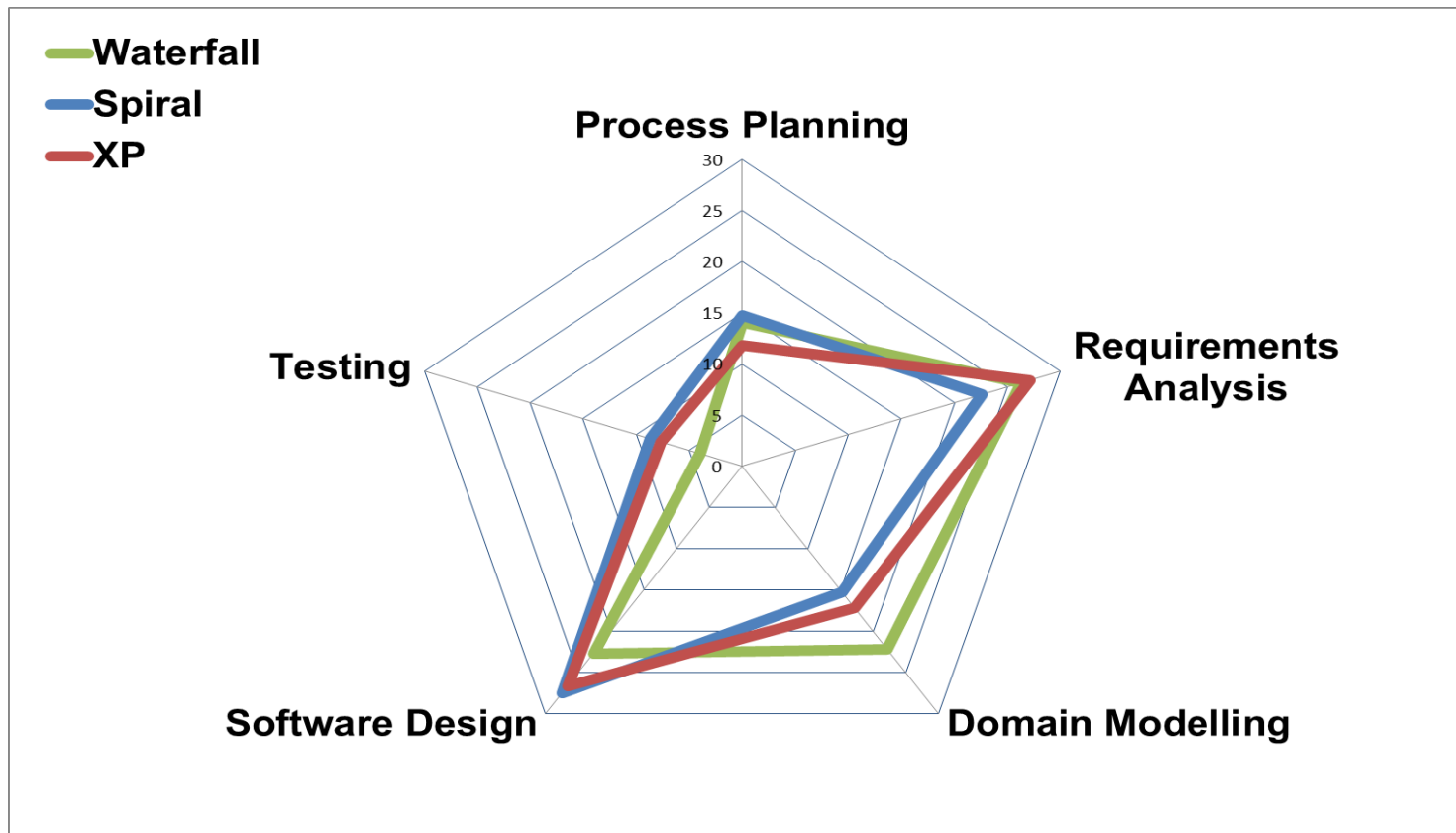
Size of documentation

	Waterfall		Spiral		XP	
	Init.	Final	Init.	Final	Init.	Final
Process Planning	8	8	7	8	6	6
Requirements Analysis	15	15	11	13	12	15
Domain Modelling	13	13	8	9	9	9
Software Design	13	13	11	15	15	15
Testing	2	2	5	5	4	4
Other	7	7	6	6	5	5
Total		58		56		54



Comparing the processes

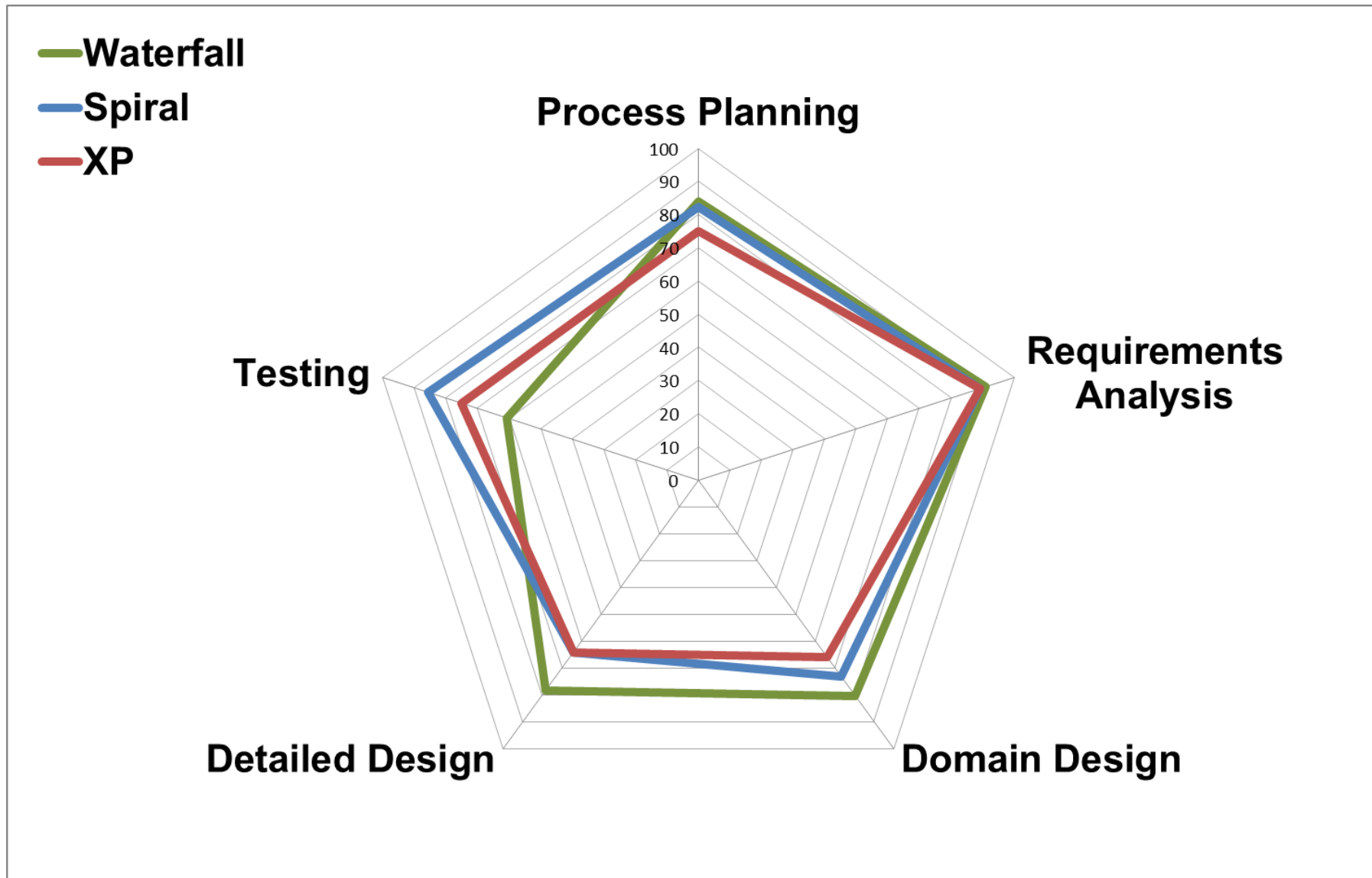
Size of documentation by phase





Comparing the processes

Quality of documentation by phase





Comparing the processes

Quality indicators

	Waterfall	Spiral	XP
Average total LOC by process model	4469	2671	5125
Ratio comments/total LOC	17%	18%	24%
Sum of cyclomatic complexity of all nested functions or methods	598	356	522
Max cyclomatic complexity	17	28	21
Faults after release by KLOC	0.96	0.90	0.84



Comparing the skills

- Three types of teams: CS only, IM only, and mixed
- Grading by the teaching assistant (coauthor Zuppiroli)
- Incentives for teams to be first to deliver



Comparing the skills

Distribution of processes by team type

	Waterfall	Spiral	XP	Total
Teams CS only	2	2	2	6
Teams IM only	2	2	3	7
Mixed teams	3	3	2	8
Total	7	7	7	21



Comparing the skills

Average marks by team type

	Waterfall	Spiral	XP	Avrg
Teams CS only	26	23	23	24
Teams IM only	22	24	22	22,5
Mixed teams	24	24	25	25,3
Total	24	23,6	23,3	21

Max grade is 30



Comparing the skills

Average marks by phase

	CS	IM	mixed
Requirements	25	27	28
Design	22	21	21
Implementation	25	24	27
Testing	26	23	24
Avrg total LOC	4938	4139	3513

Max grade is 30



Discussion

- Computer Science students got a higher score when using Waterfall
- IM students were more comfortable with the Spiral, more suitable for risk analysis
- Mixed teams scored better by using XP



Evaluation by the students

- Based on questionnaires
- 80 answers (2012), 21 answers (2011), 40 answers (2010)

Q. Are you satisfied by this course?

2010: 17% yy, 61% y, 20% n, 2% nn

2011: 20% yy, 60% y, 20% n, 0% nn

2012: 10% yy, 45% y, 25% n, 20% nn



Conclusions

- CS students were more comfortable with detailed rules as in Waterfall and more successful in coding and debugging
- IM students were more comfortable with an iterative and risk oriented process like RUP and more successful in requirement analysis
- Both type of students were poor designers, especially in using/exploiting design patterns
- XP was relatively simple to apply and most effective with mixed teams
- CS students did not like to “compete” with non-CS students



Thanks!

Questions?