SERE: Software Engineering for the Research Environment

Formal Technical Reviews

for Research Projects

int i, j, k:
while…

Lancaster University

Andre Oboler

Computing Department

Lancaster University

27^2+2^1 2

Technical Reviews

The most widely used approach in Software Development to validate the quality of a product or process [1].

"A method involving a structured encounter in which a group of technical personnel analyses an artefact according to a well-defined process" [2].

27^2+2^1 2

How it works (in industry)

1. The producer indicated they are ready for a review
2. The reviewer(s) receive the artefacts to be reviewed
3. The reviewers spend 2 hours inspecting the artefacts
4. The review takes place
    1. The producer walks through their product
    2. The reviewer(s) ask questions on the walk through
    3. The reviewers ask questions from their notes
5. A set of recommendations is produced and given to the producer to enable improvement

---

27^2+2^1 2

FTR use in academia

TABLE I Use of FTRs in Australian Sample

	Don't Know	Don't Use	Use Rarely	Use Sometimes	Use Often	Use Always
N	15	11	1	6	1	1
%	43	31	3	17	3	3

74%   6%

Taught SE: 43%

Industry Experience
<5 years: 16 (46 %)
>20 years: 6 (17 %)

TABLE II Use of FTRs in American Sample

	Don't Know	Don't Use	Use Rarely	Use Sometimes	Use Often	Use Always
N	5	2	3	6	8	1
%	20	8	12	24	32	4

28%   36%

Taught SE: 72%

Industry Experience
<5 years: 3 (12%)
>20 years: 14 (56%)

27^2+2^1 2

Comparison — Percent of Researchers vs Level of use (Don't Know, Don't Use, Use Rarely, Use Sometimes, Use often, Use always). Legend: Australian %, USA %.
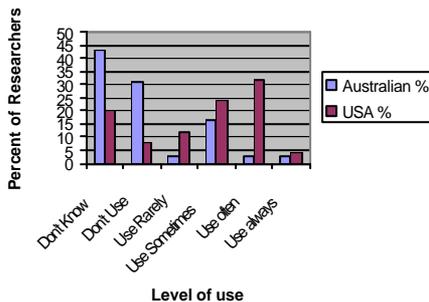
27^2+2^1 2

Why Software Engineering (inc. FTR) is not used by academic

Reason	Percent that agree with this reason
Never thought about it	14%
Don't know about them	11%
Cost of learning them is too high	17%
Not appropriate for my work	83%
Cost of use is higher than pay off	46%
Organisational Policy against spending time on them	3%

S E Software Engineering *for the* Research Environment **R E**

The Experiment

Research Aims

- To introduce Software Engineering that is appropriate to the research environment

- To reduce the costs and allow people to judge the cost/benefit better themselves

- To evaluate our interventions

---

Experimental Aims

Our approaches aimed to improve knowledge capture and facilitate greater communication

Technical Reviews were one of many Software Engineering approaches used.

Technical Reviews complimented code documentation guidelines and personal process models. Documents based on both of these were reviewed.

Engineer Assisting Researchers
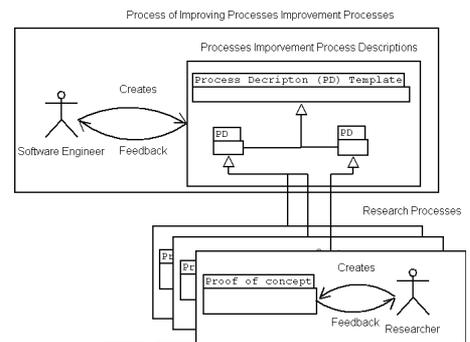
Participation

	Participants		Non Participant
Year One	Exp. 7	Control 7	15
Year Two	12		13
Year Three	11		14

Table III Marks 04-06

	Mean Project	STDEV Project	Mean Course Work	STDEV Course work	?	STDEV (?)
Participants	65.23	8.58	60.19	6.89	5.04	5.60
Non Participants	62.91	10.26	61.47	6.13	1.44	8.26
Change	2.32	-1.67	-1.28	0.76	3.60	-2.66

Back to FTRs

- Despite these positive results, only four of participants actually took technical reviews.
- The average delta for the four review participants was 4.7 (cf. 5.04 for all participants) – it is not particularly different
- Instead we must reply on feedback from the participants.

Usefulness

Participants were asked to rate the various tools (16 in year two, 19 in year three) on the following scale:

1 useless, 2 some use, 3 useful, 4 very useful, 5 the most useful

Only three of the four FTR participants returned their post project surveys. Two of these rated FTRs a 5. The remained rated them a four. This shows a perception of a comparatively high level of usefulness of FTRs.

Qualitative data

- We also recorded various interviews, FTRs, and meetings.
- The audio was transcribed and entered into NVivo where quotes were tagged.
- We have extracted and present here some of the quotes on FTRs.

Case 0417

"I haven't applied your coding practice yet and I still have a fair amount to write and already have a fair chunk of code so it could take me a while, I will get back to you when I have gone through it all and arrange a meeting!" - Student

- The student did not follow up, and after the project commented that their project suffered from a lack of documentation making it hard for others to reuse their work.

Case 0526

"It was a good work through. You did quite a lot of going through his problems, and came up with solutions… over all very informative" [to the student]" – Observer (a PhD student)

Case 0526

"the good thing is that in the end you have pretty good documented code" - Student
After a 1.5 hour FTR student asked if it was worthwhile: "Of course, yeah, yeah of course. I really like the dOxygen, it seems really nice… the ideal thing would be for me to have all of this prepared, all the documentation, but it's quite a task."

Case 0610

"The clarification of how I use this with my supervisor answered a lot of questions, basically he asked a lot of questions that I would have asked, your comments were useful." – Student

"Interesting, it was good to be here… to be brutally honest I'm not convinced how useful the technology is, I think the most useful thing they'll get out of it is simply another point of view… You're further way from the project and have a different perspective and that's quite useful."   - Student's supervisor

Case 0628

"From the perspective of reusing the project later on the reviews were useful" – Student

"If I come back 18 months, or even a year later and want to understand what that's talking about... for example going through a technical review to find out the problems before I archive it, would be of enormous benefit." – Student on potential use in a longer term project

Case 0525

But it's not for everyone…one non-participant was asked if they'd participate with hindsight:

"No, it sounds like something that would complicate work. I prefer to just do my own thing even if it might take longer - it's probably just a reluctance to use other people's tools". – Student

Other benefits

- Our use of technical reviews for students process plans proved very successful
- The artifact was in this case only vaguely related to the computer, but was structured
- FTRs could be systematically applied to other parts of the research process
- Configuration settings and other computer aided tasks where design decisions are made would be a good target for further investigation

Summary

- FTRs allows a systematic approach reviewing and suggesting improvements to artifacts
- Widely used outside academia in the Software Development field, they are not extensively used inside academia, even in computer science departments
- FTRs are of benefit to research projects
- They may have be adaptable to other uses in the research environment, particularly those involving computers where decisions are made on configuration settings

Thank you

Questions?

References

1. Sommerville, I., *Software Engineering, 6th edition*. 2001, Harlow, UK: Addison-Wesley.
2. Johnson, P., *The WWW Formal Technical Review Archive*. 1999, University of Hawaii.