

Scientific Literature (@ TUM)

Dr. Benedikt Hauptmann

Dr. Henning Femmer

With material from

Dr. Florian Deißeböck, and

Prof. Dr. Stefan Wagner

Goals and Content

1. Methodology: Searching for literature in a scientific field
2. Evaluation: Indicators for quality of scientific papers
3. Technical Aspects (@TUM): Getting a paper for a citation

Two parts:

1. What are scientific publications?
2. Literature search

Why literature reviews?

- Part of your job.

Goals of literature reviews:

- Understand the state of science
- Identify currently open questions
- Show relevance
- Define commonalities and differences with other work (and explain why)
- Place your work in the area of research
- Give evidence for your assumptions
- ...

Why?



Scientific Publications

(a tiny introduction)

What kind of paper/articles/... exist?

Research type facets [Wieringa2005]

- Exploratory
- Solution
- Validation / Evaluation
- Experience
- Philosophical / Opinion

[Wieringa2005] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, “*Requirements engineering paper classification and evaluation criteria: a proposal and a discussion*”, Req. Eng., 2005.

What kind of paper/articles/... exist?

Jackpot: Meta publications / Research Surveys

Conference on Systems Engineering Research (CSER'13)

Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

A Literature Survey on International Standards for Systems Requirements Engineering

Florian Schneider^{a*}, Brian Berenbach^b

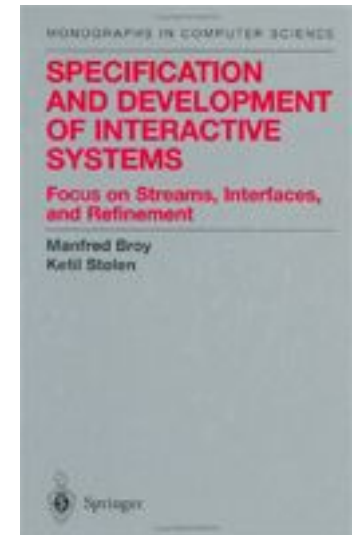
^a*Chair for Applied Software Engineering, Technische Universität München, Boltzmannstr. 3, Garching, 85748, Germany*
^b*Siemens Corporation, Corporate Technology, 755 College Road East, Princeton 08540, USA*

Which formats?

**Different Formats
-> Different Quality**

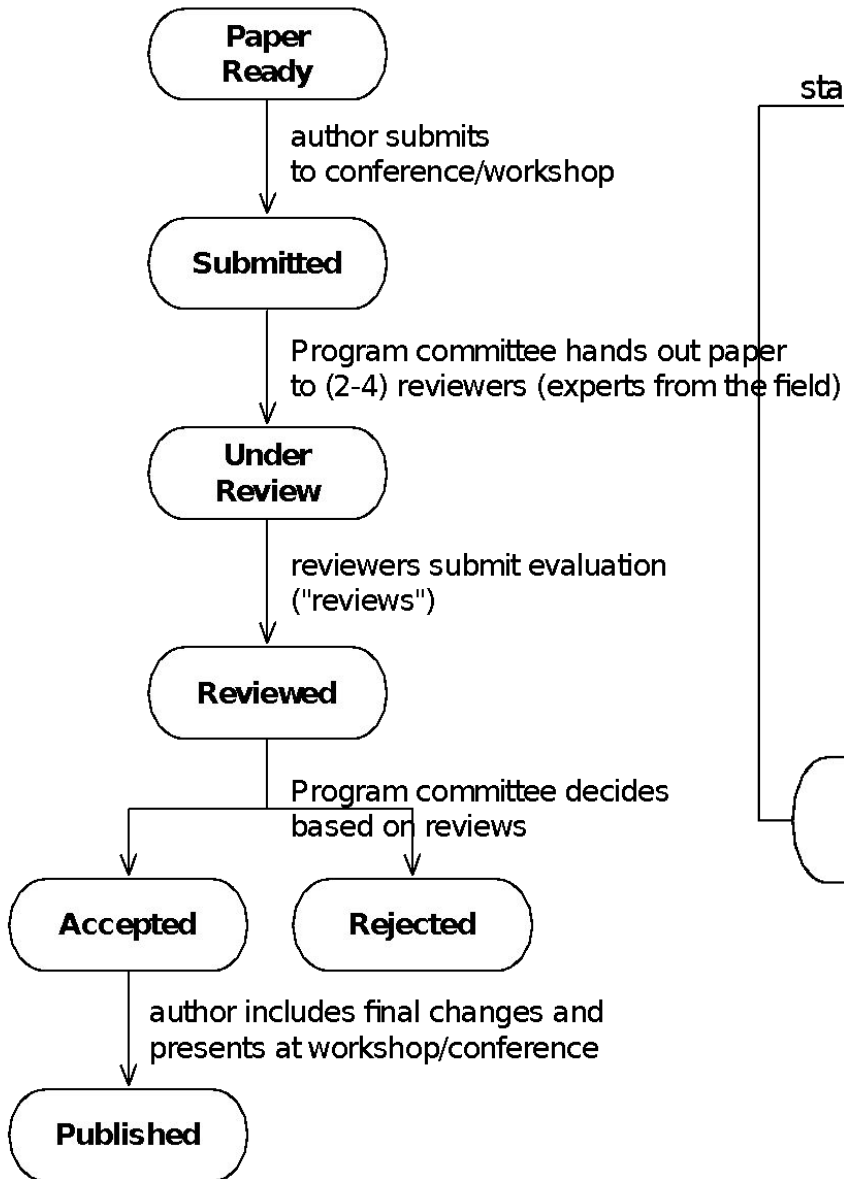
**It's all about the
peer review**

- **Book**
 - Usually single-author
 - 100 – 1000 pages
- **Book chapter**
 - 20 – 50 pages
- **Journal article/paper**
 - 10 – 30 pages
- **Proceedings article/paper (conference)**
 - 3 – 15 pages
- **Workshop article/paper**
 - 3 – 15 pages
- **Technical reports**
- **Thesis (Dissertation/Master's Thesis/Bachelor's Thesis)**
- **Blogs**
- **Tweets**
- ...

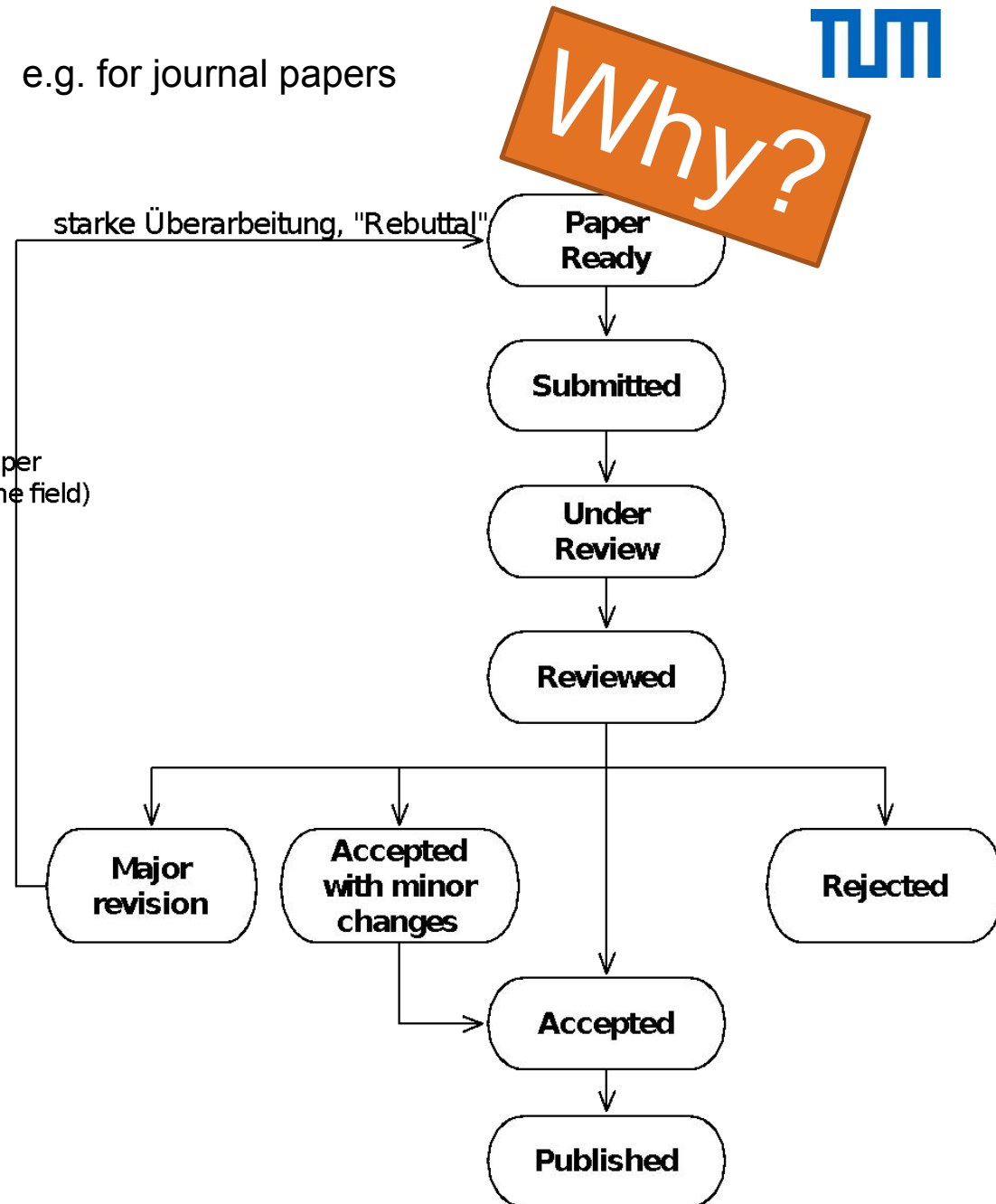


The Peer Review System

e.g. for conferences



e.g. for journal papers



Why?

----- REVIEW 2 -----

PAPER: 4

TITLE: Who is the Advocate? Stakeholders for Sustainability

AUTHORS: Birgit Penzenstadler, Henning Femmer and Debra Richardson

----- REVIEW -----

The paper attempts to devise a systematic process for identifying advocates of sustainability. The process leverages on well-established/classical requirements engineering processes for stakeholders identification.

- + The paper is well-written and presented. The examples are useful.
- + The use of the requirements techniques to identifying sustainability stakeholders looks to be plausible.
- Combining/contrasting the outcomes the four techniques look to be an expensive exercise. Furthermore, sustainability analysis may also require the identification of an equally important slice of stakeholders, who are the devils' advocates for sustainability. This could be important for promoting behavioral change, analyzing obstacles for the sustainability agenda, understanding their incentives, quantifying the risks to adoption and working on solutions for promoting acceptance.
- I would also expect more justification of the exercise in terms of long term benefits and influences on various software artifacts. The motivation is bit weak to justify the effort.

----- REVIEW 1 -----

PAPER: 8

TITLE: Detecting Inconsistencies in Wrappers - A Case Study

AUTHORS: Henning Femmer, Dharmalingam Ganesan, Mika

= Overall opinion =

It's a pleasure to read such a self-contained paper that answers pretty much every question the reader raises while reading. It's well structured and provides both detailed background theory and insightful

----- REVIEW 2 -----

= Paper overview =

The paper addresses the problem of detecting inconsistencies in software abstractions underlying implementations.

The authors propose a method for detecting the equivalence of several abstractions and the modification of parameters. The method is neither complete nor sound.

The solution proposed is based on static analysis and learning. The static analysis is based on a given implementation of the wrappers. The wrappers are "extractors" driving the analysis. The analysis produces differences in return values. The analysis produces some false negatives (as in the case study). These were corrected in the solution. The solution also integrates a learning phase. Finally, the method produces information of the state of the analysis. The method keeps the "best" data.

The methodology is to transform the code composed of 10 lines into a C code. It revealed 84 issues, of which 10 are failures. Each type of failure is given. The industrial team is given.

= Detailed feedback =

- abstract - typo: "instead of to the" -> "instead of the"

- p2 - typo: "an SAL" -> "a SAL"

- p3 - typo: "an analysts" -> "an analyst"

- p3: "The tool detects differences in the function pair and highlights the important ones to the user." -> at this stage we don't understand how the "important ones" are defined. It is also unclear at this stage that the training set starts empty (all functions unclassified) and is incrementally defined/increased. I would suggest to be more explicit when describing step (1) and (3).

- p3 - fig 2: the "very light-red background" is very hard to see; I would strongly recommend to find a better way to highlight.

- p3 - fig 2: why are "uint32" and "free" highlighted although they are identical in both implementations?

- p5: "304 pairwise comparisons": can't thing be optimized here? Eg: if f1 and f2 are equivalent and f2 and f3 too, we don't really need to check for f1 and f3, do we?

- p7- fig 6: I would suggest to repeat (at least in the label) which OS which side corresponds to (as in "VxWorks on the left and RTEMS on the right")

to another language?

pleasant to read, with clearly stated assumptions and commented, with very few errors used, and

of 2 technologies; it is integrated, and it avoids false positives

industrial team, from using the tools methodology is applied, with promising

the target language is of the methodology which could it be applied

Example

p1 l58

4 dimensions vs. views

Btw., why is the technical dimension important? For me it is simply a subset of the economic 'dimension'. Keeping a system sustainable from a technical perspective makes it more profitable from the economic point of view. If it is not 'technically sustainable', it is also not in the economically way.

Reviewer: 1

p1 l 44

example of green software code are mobile apps

Recommendation

Why are they green?

Comments:

#It is esse

+ 4 more pages
+ “please see the
attached file for more
detailed notes”

First some genera

p2 l4

conflicts be

Sorry to be hard c

not want to demot

?- Are reb

To me, it looks to

?- Arent th

categories to clas

I wound it confusi

How do th

I would evaluate it

Software?

might be my fault

I. Your aim is to re

p2 l10

Mine, too, I consid

What is the

Therefore, the cor

GSW?

When I think back

I think the sentence should be reformulated to avoid misunderstandings.

paper. Then I rem

----Sidebar section

the contribution of

It should be more

Sec 2 Related Work and its Quality

If one keeps in mi

p2 l22

the paper is easie

Attention in SW Engineering only recently.

evaluating the qua

- depends on whether SW E is something different than the production of software

It might still be int

distracting/annoyi

SW that is now classifiyd as greening (Green by) has been produced since many years in environmental sciences, and there has been environmental informatics for a while. There is the conference series of EnviroInfo (Environmental Informatics) (27th this year). I had a lectures how to design environmental software in 1994. Early (german) names are e.g. R Grützner, B Page, V. Wohlgemuth. At least since 2003 there is a Journal on Environmental Informatics (methodologies, applications, and policy considerations, the needs for environmental systems analysis, the challenges of environmental systems modeling, and the impacts of environmental informatics are discussed etc., btw., IF 3.619, not that I like Impact Factors).

that the classificat

One more genera

Is green software

I have the feeling

enough knowledg

- [illegible]



How *good* is a scientific publication?

Quality of scientific publications has several aspects:

- Relevance
- Novelty
- Validity
- Rigor
- ...

Tough stuff...

How *good* is a scientific publication?

Easier to judge:

1. Where was it published (venue)?

- Peer-reviewed venue
- Impact factor of venue
- Acceptance rate of venue

2. How was it received in literature?

- Number of citations

Use with care...

Engineering venues*

Guess:
Acceptance Rate

Top General SE Conferences	ICSE	FSE/ESEC	ASE	SPLASH/OOPSLA	ECOOP	ISSTA	FASE
2013	85/461(18%)	51/251(20%)	?	?	?	32/124(26%)	26/112(23%)
2012	87/408(21%)	34/201(17%)	?	?	?	31/108(29%)	33/134(25%)
2011	62/441(14%)	34/203(17%)	?	?	?	35/121(29%)	29/99(29%)
2010	52/380(14%)	34/169(20%)	37/252(15%)	61/166(37%)	?	24/105(23%)	24/96(25%)
2009	50/405(12%)	32+7/217(15%)	34+31/191(18%)	?(28%)	?(23%)	25/93(27%)	30/124(24%)
		38+33/222(17%)	25/144(17%)	25/117(21%)			

- Journal of Software Engineering
- Wiley Journal of Software Engineering
- Springer Empirical Software Engineering

Conferences (Proceedings)

- International Conference on Software Engineering (ICSE)
- Foundations of Software Engineering (FSE)
- International Conference on Automated Software Engineering (ASE)
- OO Programming, Systems, Languages and Applications (OOPSLA)
- International Symposium on Software Testing and Analysis (ISSTA)
- International Conference on Software Maintenance (ICSM)

• <https://taoxie.cs.illinois.edu/seconferences.htm>

*in no particular order

Hands-on:

Literature reviews

Where can we find papers?

Publishers:

- ACM Digital Library
- IEEE Xplore
- Springer Link
- Elsevier
- TUM library
- ...

Direct sources

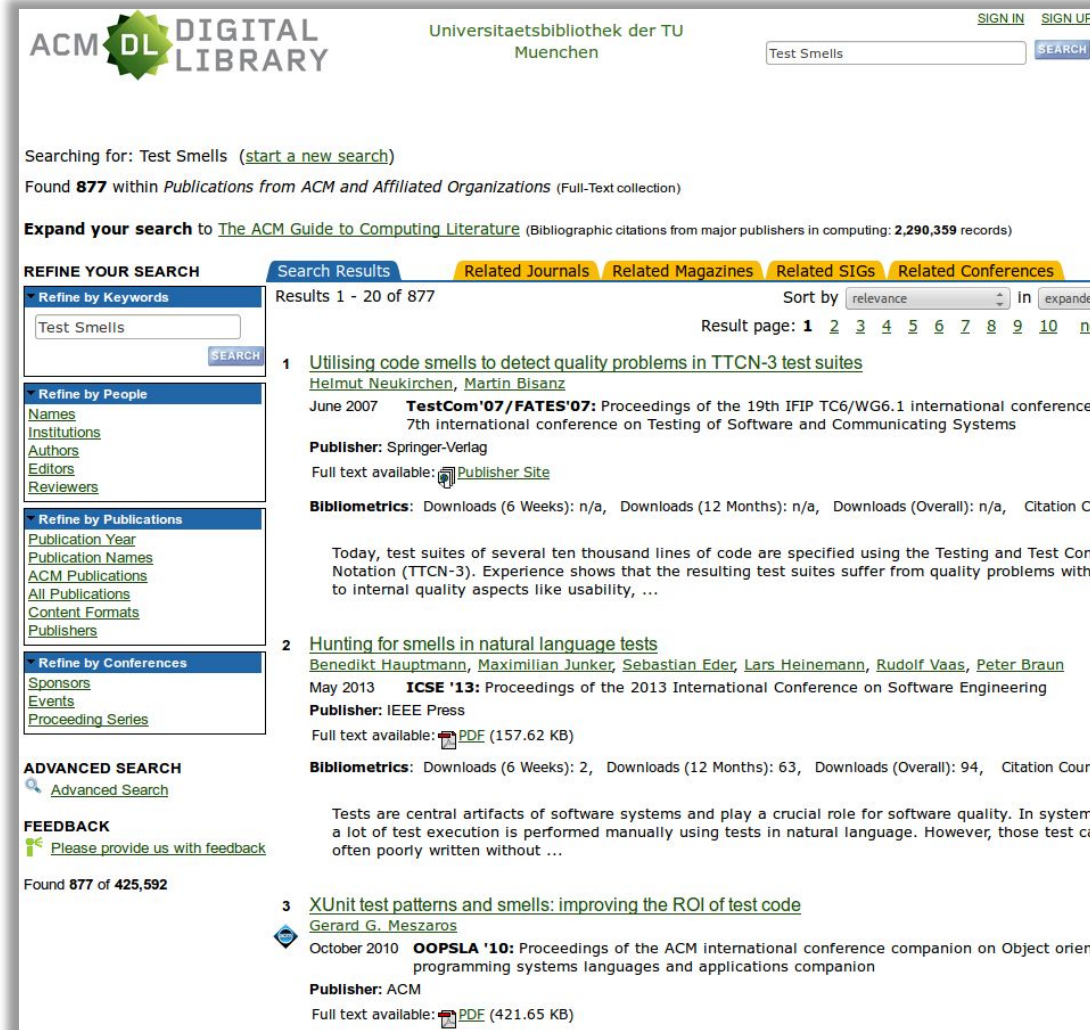
- Authors homepages

Meta sources

- scholar.google.com
- Research Gate

We use mostly Google Scholar!

- Pro
 - All results in one place
 - Direct meta-information (citations)
 - Sometimes direct link to PDF
 - Author graphs
- Con
 - No quality filter
 - Few search filter options



The screenshot shows the ACM Digital Library search results for the query "Test Smells". The page header includes the ACM Digital Library logo, the text "Universitätsbibliothek der TU München", and a search bar containing "Test Smells". Below the header, it states "Searching for: Test Smells (start a new search)" and "Found 877 within Publications from ACM and Affiliated Organizations (Full-Text collection)".

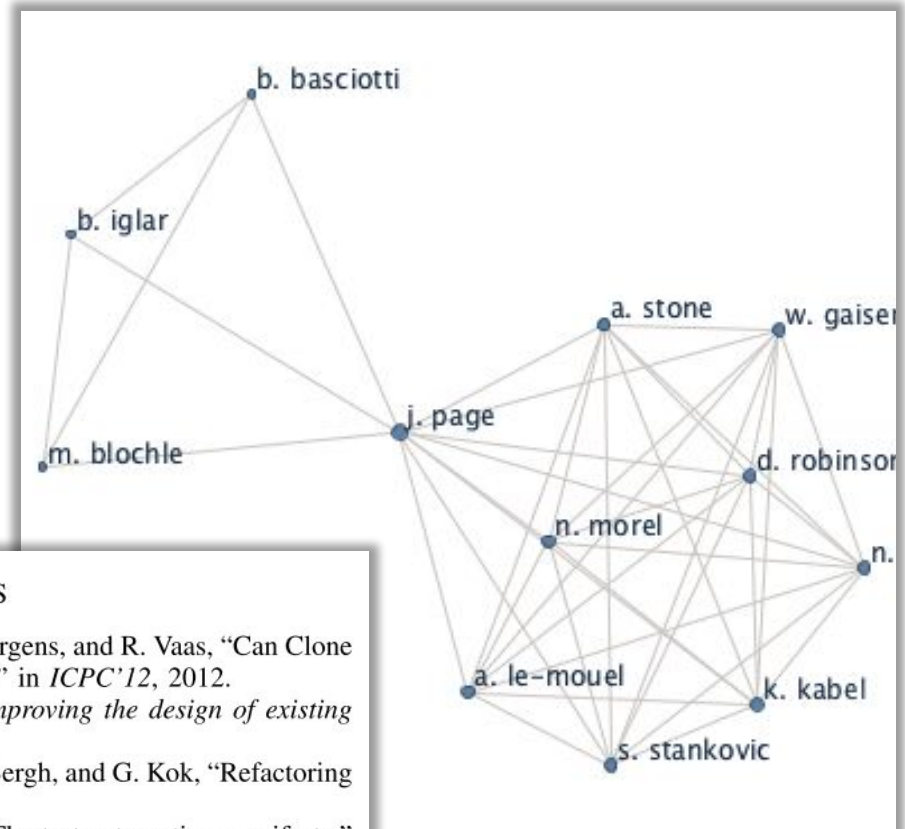
On the left side, there is a "REFINE YOUR SEARCH" section with three expandable categories: "Refine by Keywords" (containing "Test Smells"), "Refine by People" (with links for Names, Institutions, Authors, Editors, Reviewers), and "Refine by Publications" (with links for Publication Year, Publication Names, ACM Publications, All Publications, Content Formats, Publishers). Below this is an "ADVANCED SEARCH" section with a link to "Advanced Search" and a "FEEDBACK" section with a link to "Please provide us with feedback".

The main content area shows "Search Results" for "Results 1 - 20 of 877". It lists three results:

- Utilising code smells to detect quality problems in TTCN-3 test suites** by Helmut Neukirchen, Martin Bisanz. June 2007. **TestCom'07/FATES'07**: Proceedings of the 19th IFIP TC6/WG6.1 international conference on 7th international conference on Testing of Software and Communicating Systems. Publisher: Springer-Verlag. Full text available: [Publisher Site](#). **Bibliometrics**: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Downloads (Overall): n/a, Citation Count: n/a. Abstract: Today, test suites of several ten thousand lines of code are specified using the Testing and Test Case Notation (TTCN-3). Experience shows that the resulting test suites suffer from quality problems with respect to internal quality aspects like usability, ...
- Hunting for smells in natural language tests** by Benedikt Hauptmann, Maximilian Junker, Sebastian Eder, Lars Heinemann, Rudolf Vaas, Peter Braun. May 2013. **ICSE '13**: Proceedings of the 2013 International Conference on Software Engineering. Publisher: IEEE Press. Full text available: [PDF](#) (157.62 KB). **Bibliometrics**: Downloads (6 Weeks): 2, Downloads (12 Months): 63, Downloads (Overall): 94, Citation Count: n/a. Abstract: Tests are central artifacts of software systems and play a crucial role for software quality. In system testing, a lot of test execution is performed manually using tests in natural language. However, those test cases are often poorly written without ...
- XUnit test patterns and smells: improving the ROI of test code** by Gerard G. Meszaros. October 2010. **OOPSLA '10**: Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion. Publisher: ACM. Full text available: [PDF](#) (421.65 KB).

Search strategies

1. Manually searching for keywords
2. Searching through author's pages
3. Literature Snowballing
4. Systematic strategies
 - Systematic Mapping Studies
 - Systematic Literature Reviews



REFERENCES

- [1] B. Hauptmann, M. Junker, S. Eder, E. Juergens, and R. Vaas, "Can Clone Detection Support Test Comprehension?" in *ICPC'12*, 2012.
- [2] M. Fowler and K. Beck, *Refactoring: improving the design of existing code*. Addison-Wesley, 1999.
- [3] A. van Deursen, L. Moonen, A. van den Bergh, and G. Kok, "Refactoring Test Code," in *XP'01*, 2001.
- [4] G. Meszaros, S. Smith, and J. Andrea, "The test automation manifesto," in *XP'03*, 2003.
- [5] G. Meszaros, *xUnit Test Patterns: Refactoring Test Code*. Addison-Wesley, 2007.
- [6] M. Abbes, F. Khomh, Y.-G. Gue andhe andneuc, and G. Antoniol, "An empirical study of the impact of two antipatterns, blob and spaghetti code, on program comprehension," in *CSMR'11*, 2011.
- [7] F. Khomh, M. Di Penta, and Y.-G. Gueheneuc, "An exploratory study of the impact of code smells on software change-proneness," in *WCRE'09*, 2009.
- [8] A. van Deursen and M. Leon, "The video store revisited thoughts on refactoring and testing" in *XP'02*, 2002.

I know a paper. How do I get the pdf?

- Papers are licensed by the publishers (ACM, IEEE, Springer, Elsevier, ...)
- TUM has bought *most* of the licenses

How do you get access?

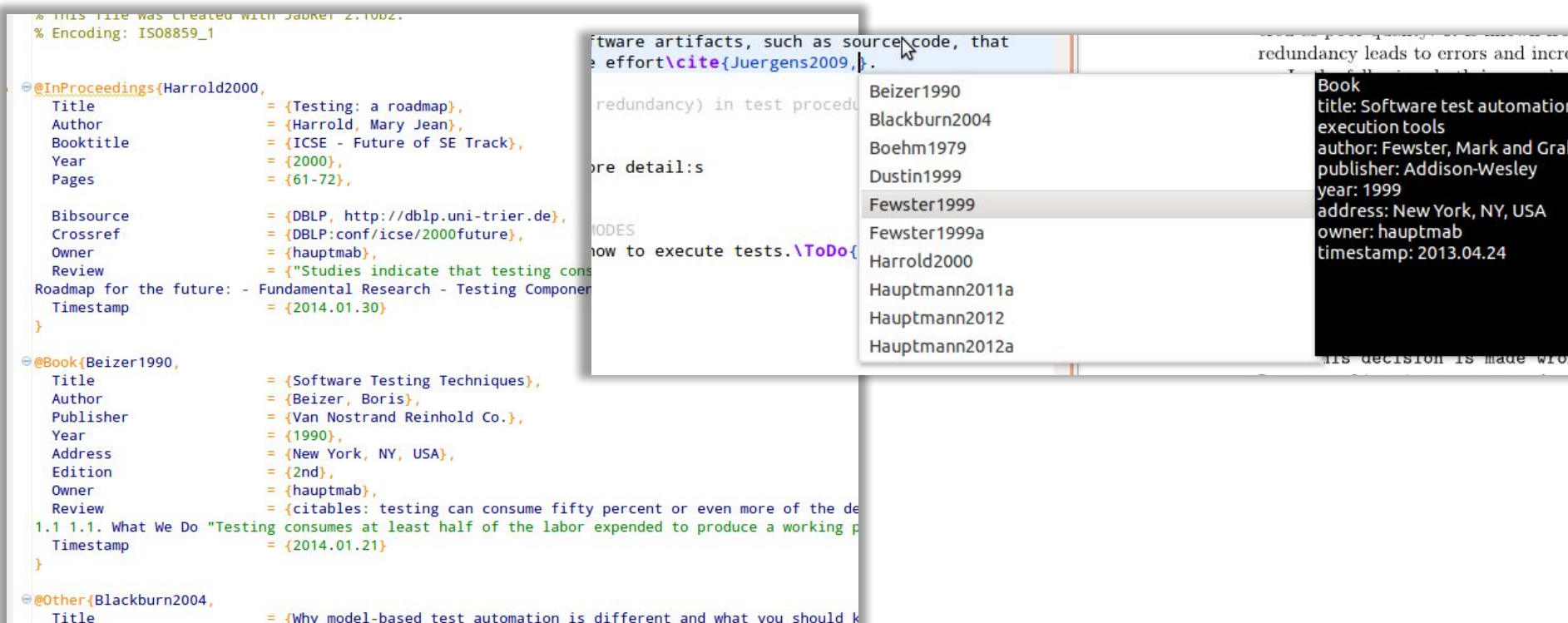
- Library computers
- <https://EACCESS.ub.tum.de> and search for venue/journal
- VPN + Proxy server

Details can be found here:

<https://www.lrz.de/services/netzdienste/proxy/journals-access/>

Reference management

- BibTeX (+LaTeX)
 - Classical, useful format for LaTeX
 - Most sources offer information in bibtex format
 - Use proper types: @article, ...



The screenshot shows a LaTeX source file with BibTeX entries. A context menu is open over the citation `\cite{Juergens2009}`. The menu lists several references, with `Fewster1999` highlighted. A separate window displays the full details for the selected reference.

BibTeX Entries:

```

% This file was created with JabRef 2.10b2.
% Encoding: ISO8859_1

@InProceedings{Harrold2000,
  Title           = {Testing: a roadmap},
  Author          = {Harrold, Mary Jean},
  Booktitle       = {ICSE - Future of SE Track},
  Year            = {2000},
  Pages           = {61-72},

  Bibsource       = {DBLP, http://dblp.uni-trier.de},
  Crossref        = {DBLP:conf/icse/2000future},
  Owner           = {hauptmab},
  Review          = {"Studies indicate that testing consumes at least half of the labor expended to produce a working program."},
  Roadmap for the future: - Fundamental Research - Testing Component
  Timestamp       = {2014.01.30}
}

@Book{Beizer1990,
  Title           = {Software Testing Techniques},
  Author          = {Beizer, Boris},
  Publisher       = {Van Nostrand Reinhold Co.},
  Year            = {1990},
  Address         = {New York, NY, USA},
  Edition         = {2nd},
  Owner           = {hauptmab},
  Review          = {citables: testing can consume fifty percent or even more of the development effort},
  Timestamp       = {2014.01.21}
}

@Other{Blackburn2004,
  Title           = {Why model-based test automation is different and what you should know}
}

```

Context Menu Options:

- Beizer1990
- Blackburn2004
- Boehm1979
- Dustin1999
- Fewster1999**
- Fewster1999a
- Harrold2000
- Hauptmann2011a
- Hauptmann2012
- Hauptmann2012a

Reference Details (for Fewster1999):

- Book
- title: Software test automation
- execution tools
- author: Fewster, Mark and Gra
- publisher: Addison-Wesley
- year: 1999
- address: New York, NY, USA
- owner: hauptmab
- timestamp: 2013.04.24

Reference management

- BibTeX (+LaTeX)
 - Classical, useful format for LaTeX
 - Most sources offer information in bibtex format
 - Use proper types: `@article, ...`
- JabRef
 - Plattform independant
 - Uses BibTeX as data format
 - Link to pdf
 - Group
- Mendeley
 - Modern, “in the cloud”
 - Cooperative references
 - Notes, highlighting, etc
 - Bibtex export
- Papers (Mac only)
- ...



Goals and Content

1. Methodology: Searching for literature in a scientific field
2. Evaluation: Indicators for quality of scientific papers
3. Technical Aspects (@TUM): Getting a paper for a citation

Two parts:

1. What are scientific publications?
2. Literature search

More details

- B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” 2007.
- S. Keshav, “How to Read a Paper”, 2013
<http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf>
- Kent Beck: How to get a paper accepted at OOPSLA:
<http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html>
- A. Zeller and T. Zimmermann, “Failure is a Four-Letter Word – A Parody in Empirical Research”
and the corresponding presentation:
<https://www.youtube.com/watch?v=NM3CIIbuVoM>
- Zugang zu Wissenschaftlichen Publikationen für Mitarbeiter und Studierende der TUM
<https://www.lrz.de/services/netzdienste/proxy/zeitschriftenzugang/>
- Paper-Verwaltung:
<http://www.mendeley.com>
<http://jabref.sourceforge.net>

...