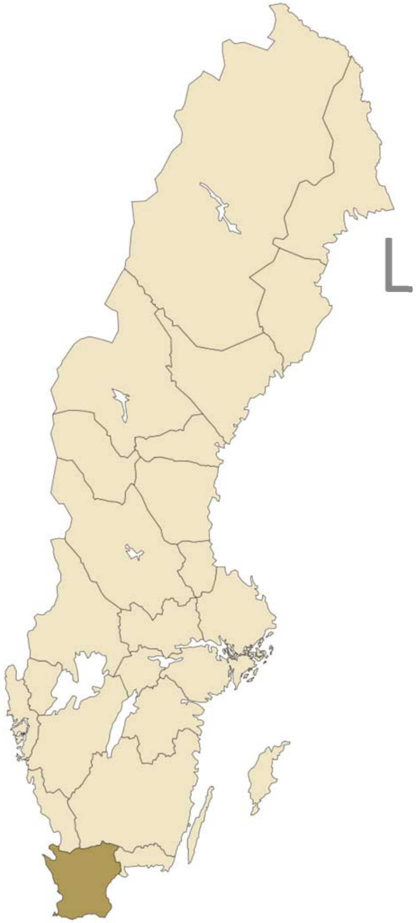
Power of the crowds and on the division of labor in software testing



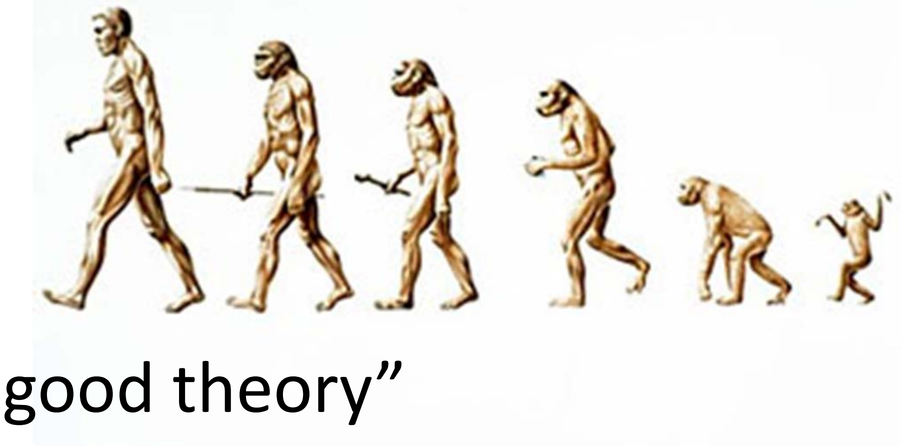
Mika Mäntylä

Lund University / Aalto University

<mika.mantyla@cs.lth.se>

08/2012->

Outline - Power of the crowds…



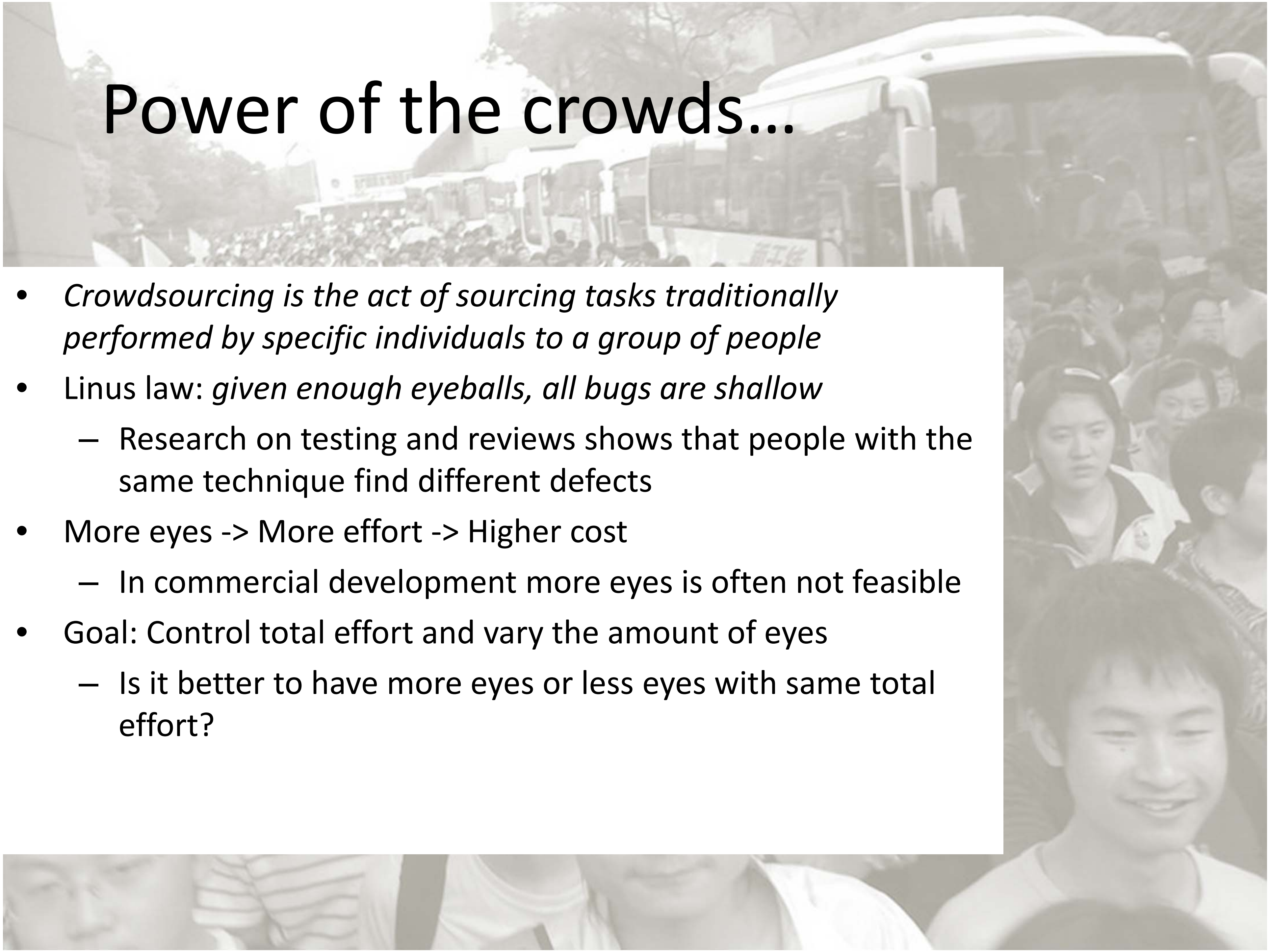
- Theoretical background
  - “Nothing is as practical as a good theory”

    Lewin K.

- Examples and evidence

- Time restricted crowds in testing

- Downsides of crowds



Power of the crowds…

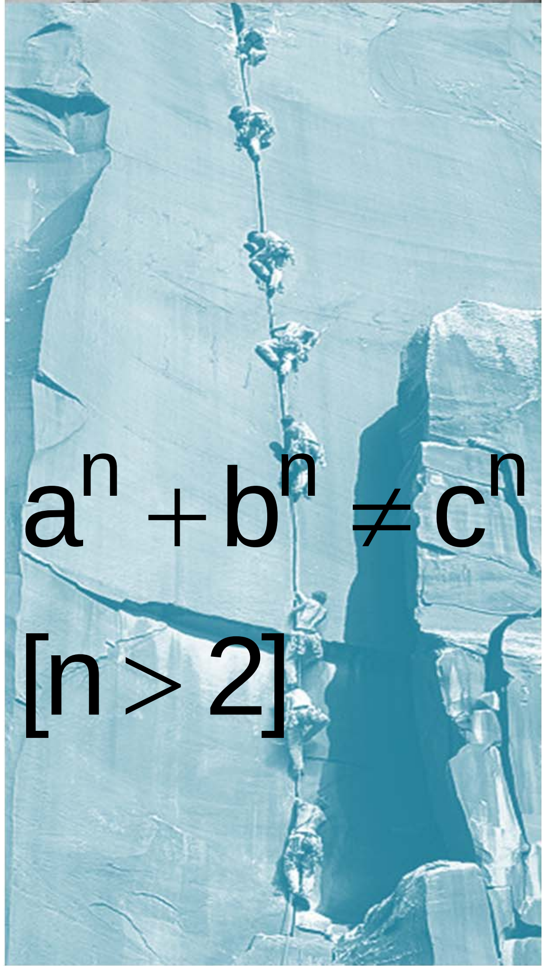
- *Crowdsourcing is the act of sourcing tasks traditionally performed by specific individuals to a group of people*
- Linus law: *given enough eyeballs, all bugs are shallow*
  - Research on testing and reviews shows that people with the same technique find different defects
- More eyes -> More effort -> Higher cost
  - In commercial development more eyes is often not feasible
- Goal: Control total effort and vary the amount of eyes
  - Is it better to have more eyes or less eyes with same total effort?

…on the division of labor in software testing

- studying group performance is meaningless unless the task type is known
- Steiner's (1972) taxonomy of tasks
  - Divisible - Unitary ~ How easily the task can be divided
    - Size matters -> Smaller tasks are often unitary undividable
    - Yet large task can be unitary,
      - e.g. reading War and Peace by Tolstoi (1475 pages)
  - Maximizing - Optimizing ~ Many items or one item with the highest quality?
    - E.g. One great phone model or several OK

…on the division of labor in software testing cont'd

- Combinability dimension in Steiner's taxonomy of tasks
  - Additive
    - Efforts are added up, e.g. pushing a car
    - Group performance is sum
  - Conjunctive
    - Every group member must perform, e.g. mountain climbing team
    - The weakest member determines group performance
  - Disjunctive
    - Only one group member must perform, e.g. coming up with right math answer
    - The best member determines group performance
- Programming (small task) is *disjunctive* and *optimizing*
- Testing (small task) is *additive* and *maximizing*

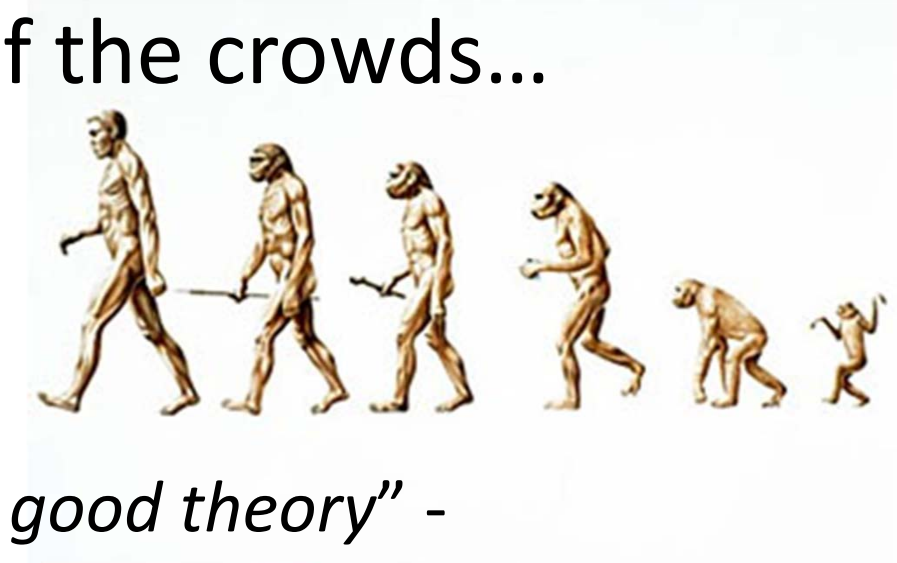


$$a^n + b^n \neq c^n$$

$$[n > 2]$$

Outline - Power of the crowds…

- Theoretical background
  - *“Nothing is as practical as a good theory”* -



- **Examples** and evidence

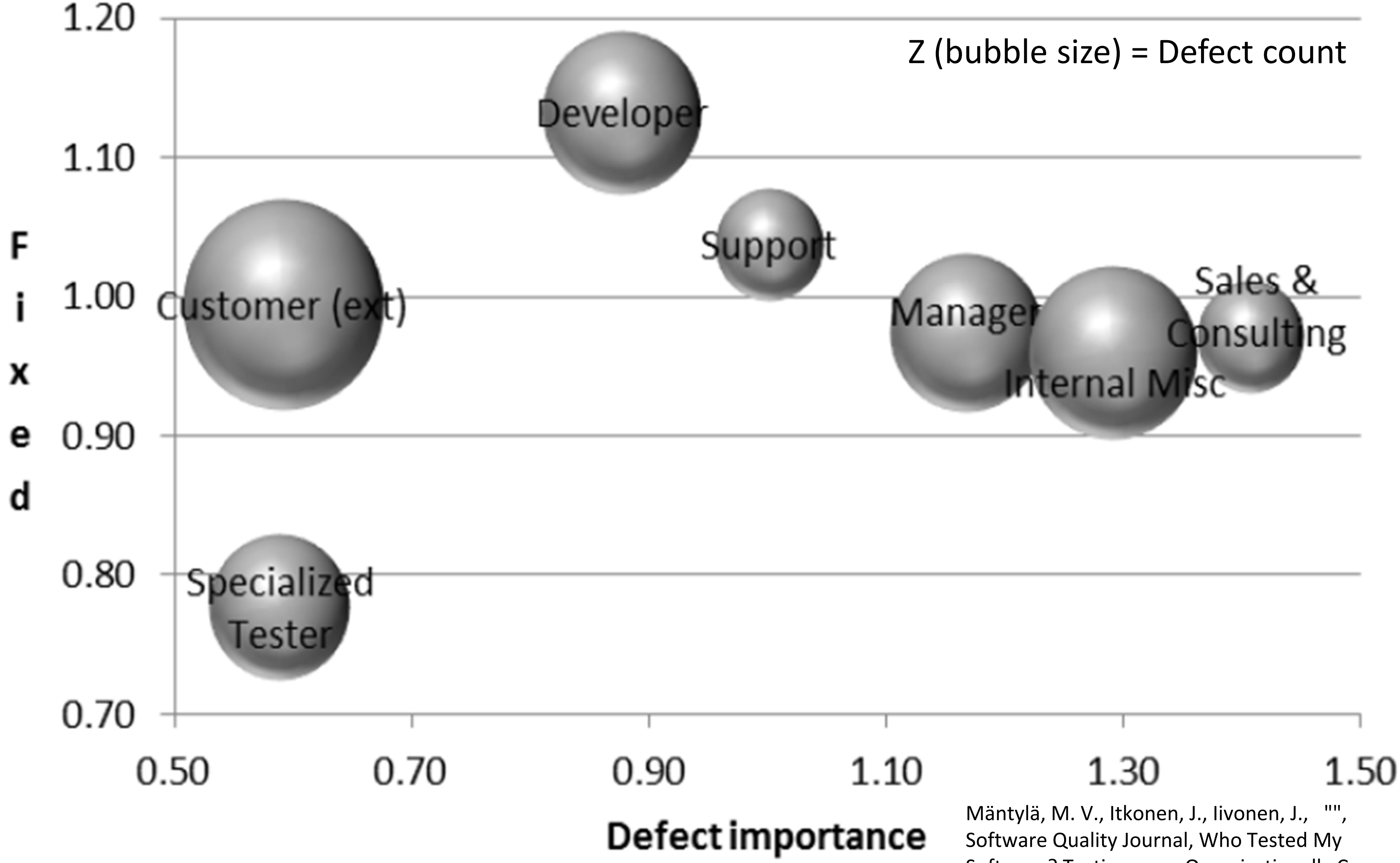
- Time restricted crowds in testing

- Downsides of crowds

Examples on how to increase the number of testers

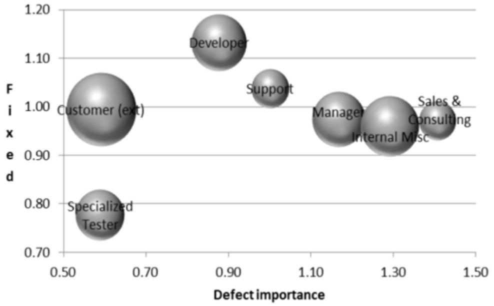
- Who tested my software
  - Also no specialist can make a contributions in testing

Who tested my software : Defect data from 3 software product companies



Mäntylä, M. V., Itkonen, J., Iivonen, J., "", Software Quality Journal, Who Tested My Software? Testing as an Organizationally Cross-Cutting Activity l

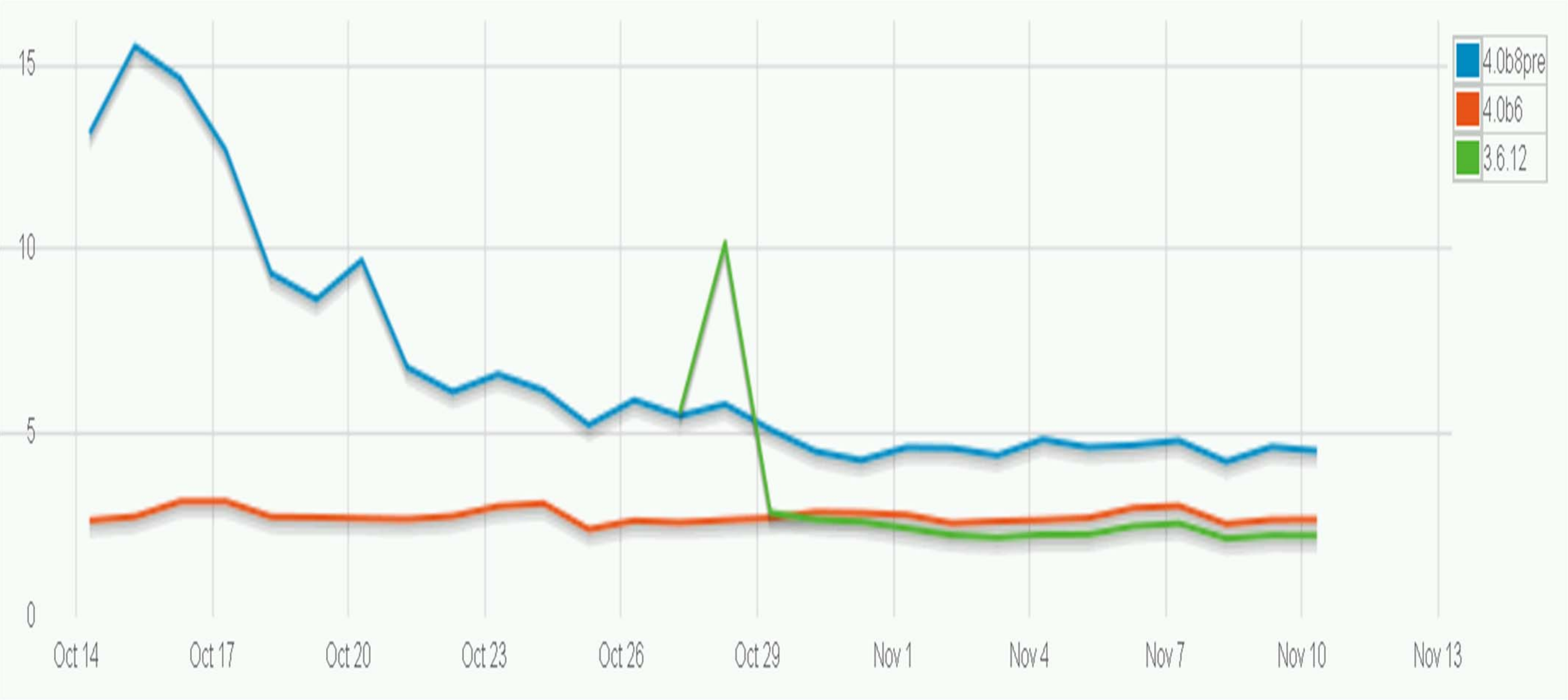
Examples on how to increase the number of testers



- Who tested my software
  - – Also no specialist can make a contributions in testing
- Internal usage of alpha/beta version of software
  - – Eating your own dog food at Microsoft
- Collecting user data (Mozilla Firefox, etc) -> make every user a tester
  - – Base product decision on data, not on opinion or politics
    - Google field tested 1000 variants of blue to figure out the correct one to use in add links (Bosch)

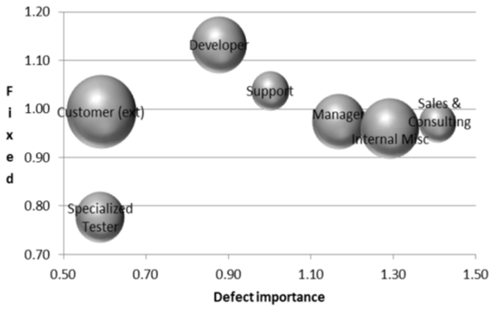
Automatic 24/7/365 quality tracking

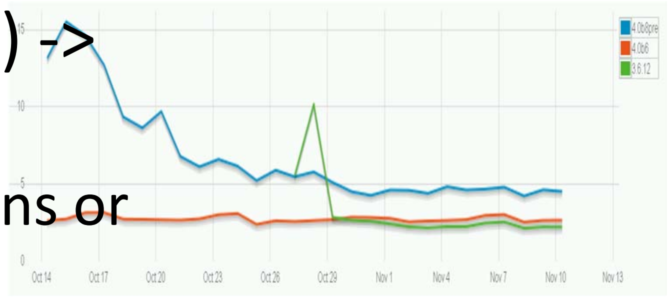
- Crash data of Firefox browser (per active daily user)
  - Blue = pre-beta
  - Red = beta
  - Green = stable



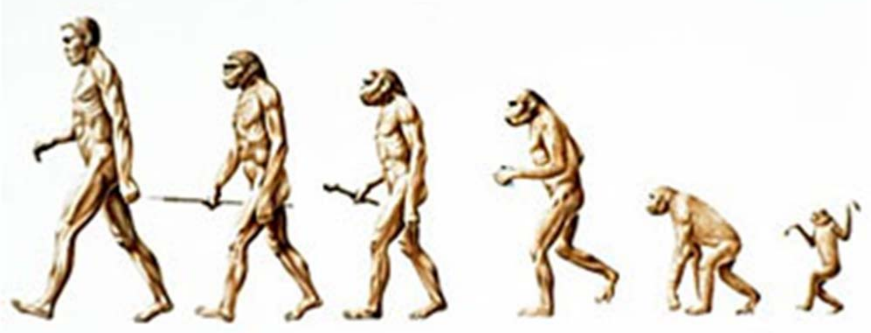
Examples on how to increase the number of testers

- Who tested my software
  - Also no specialist can make a contributions in testing
- Internal usage of the software
  - Eating your own dog food at Microsoft
- Collecting user data (Mozilla Firefox, etc) -> make every user a tester
  - Base product decision on data, not opinions or politics
- Beta testing is the most effective quality assurance method with high number of beta testers (>1000) (Jones 1996)
- Hire testers online "Just-in-Time", e.g. utest.com





Outline - Power of the crowds…



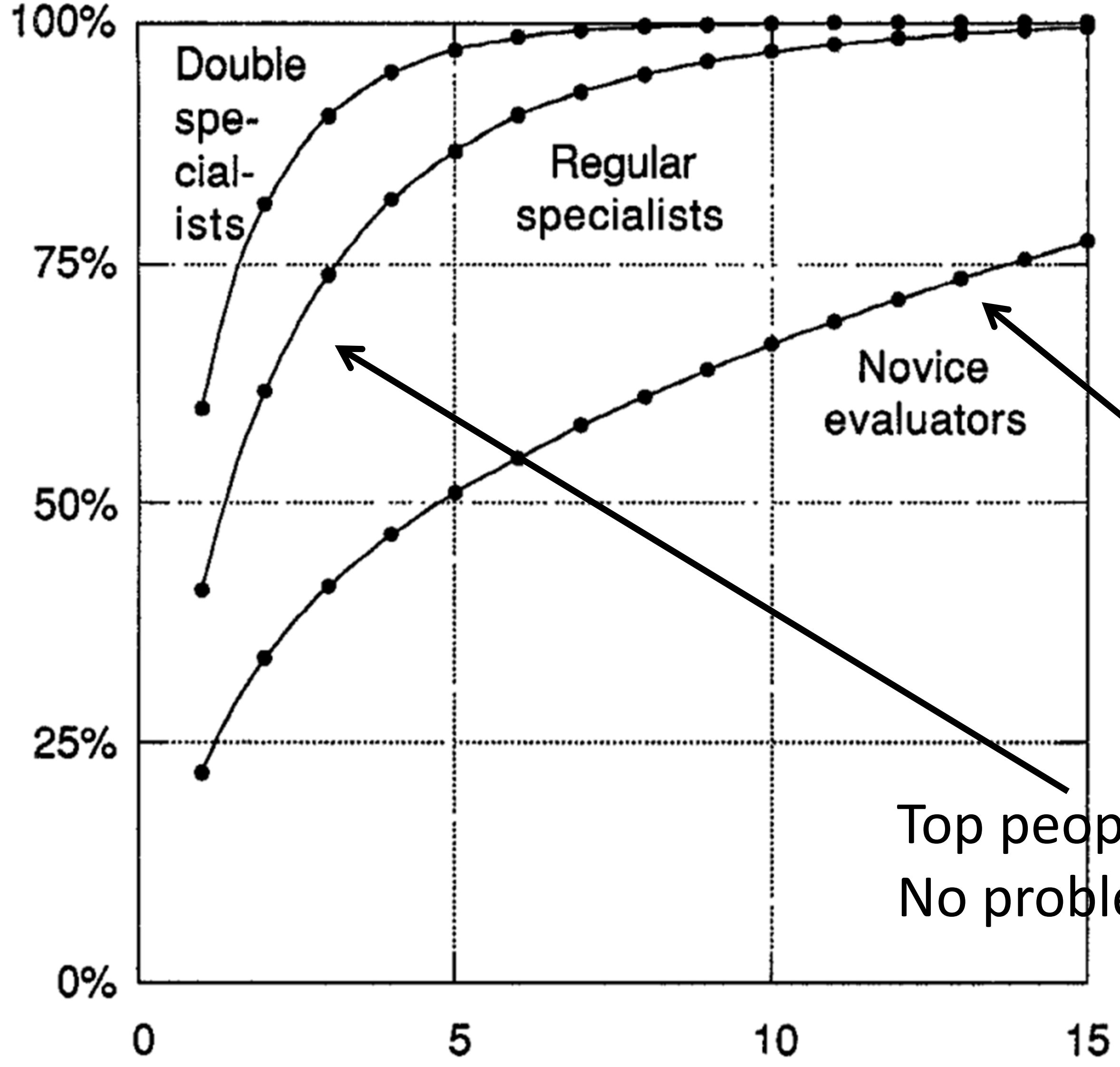
- Theoretical background
  - “Nothing is as practical as a good theory”

- Examples and **evidence**

- Time restricted crowds in testing

- Downsides of crowds

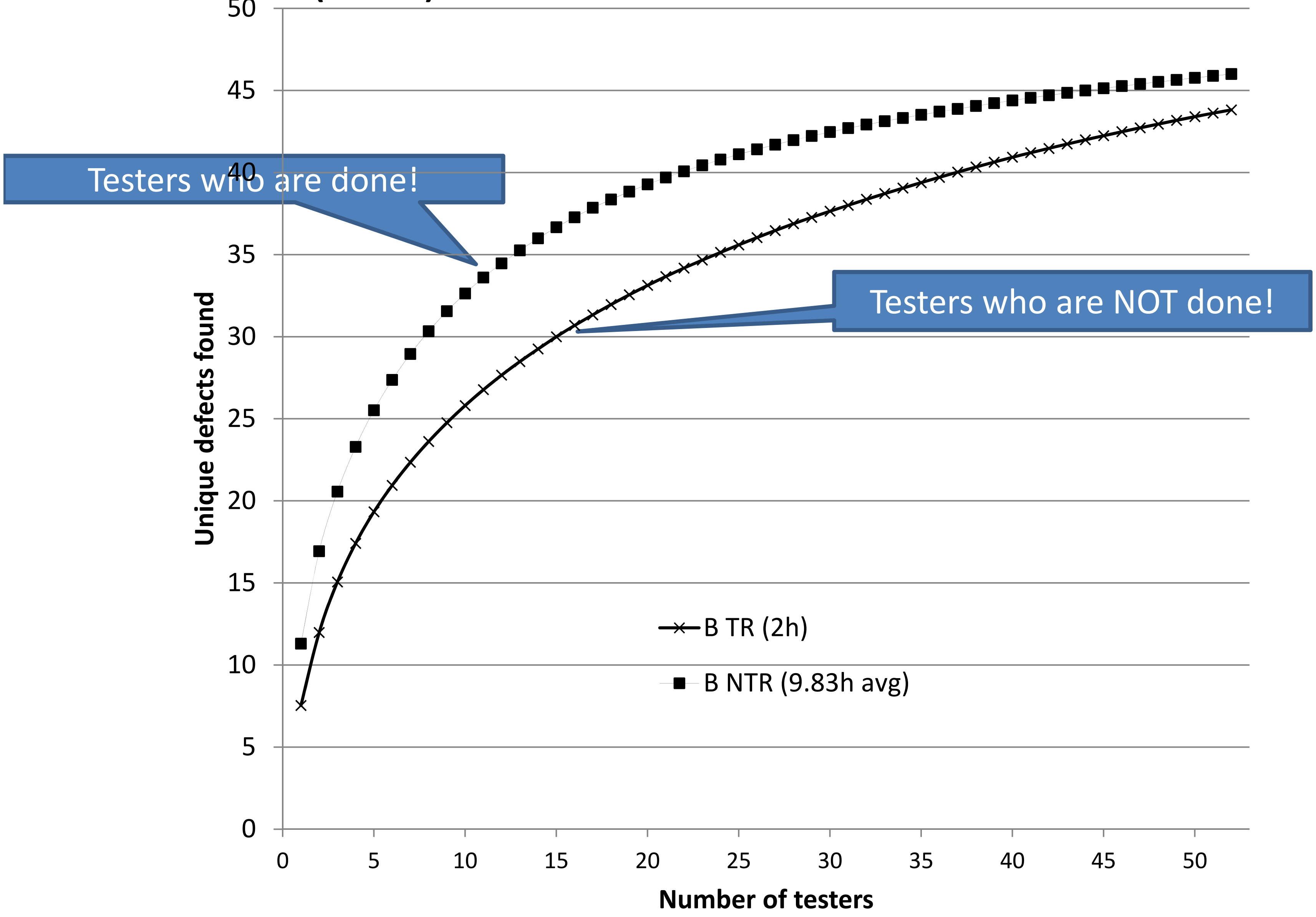
Usability: Heuristic evaluation – The effect of knowledge and share of problems detected



Top people are hard to get
No problem just use several novices

J Nielsen, ” Finding usability problems through heuristic evaluation”, 1992

Testing: addition of time restricted (TR) and non time restricted (NTR) testers leads addition in detected defects

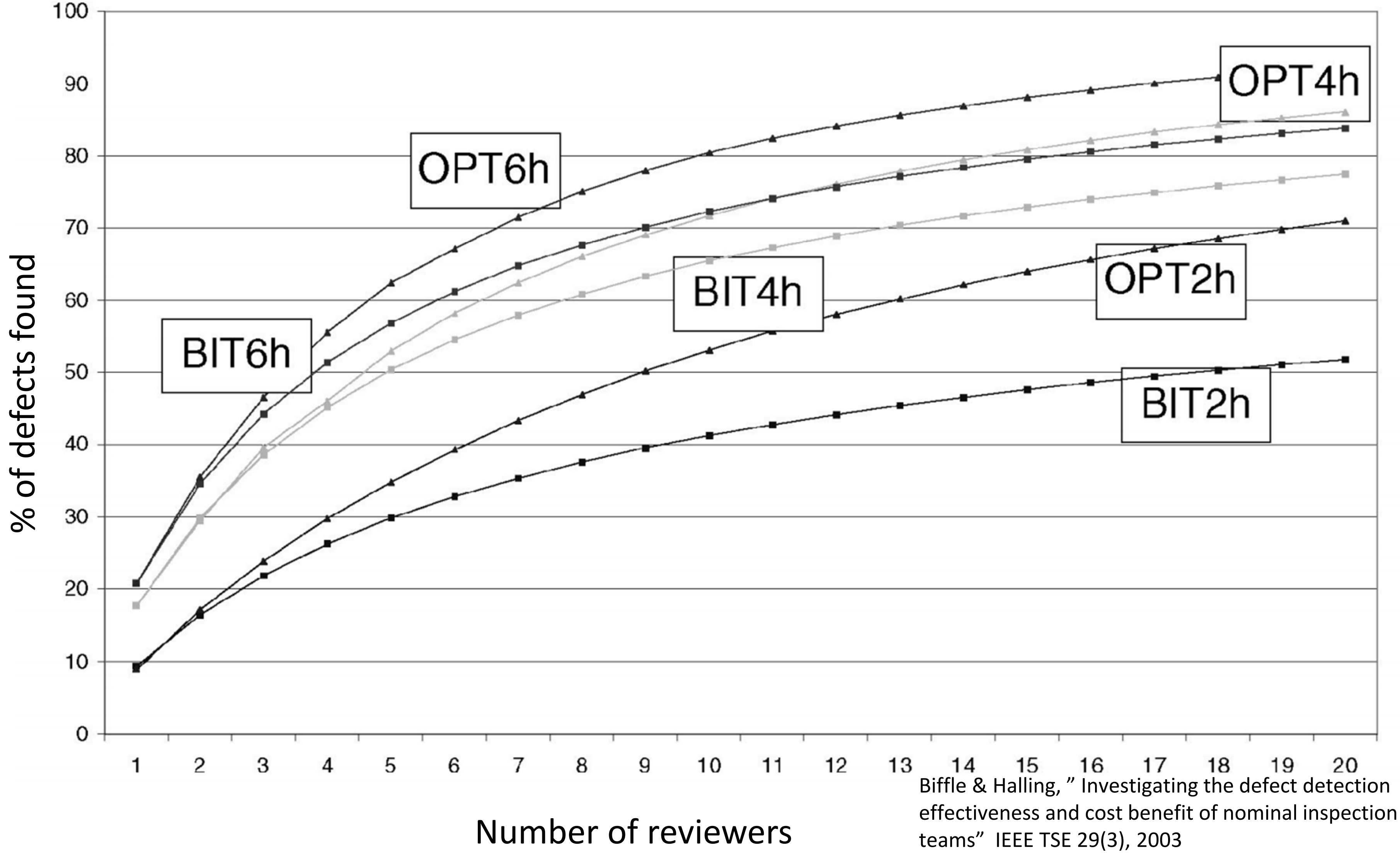


Testers who are done!

Testers who are NOT done!

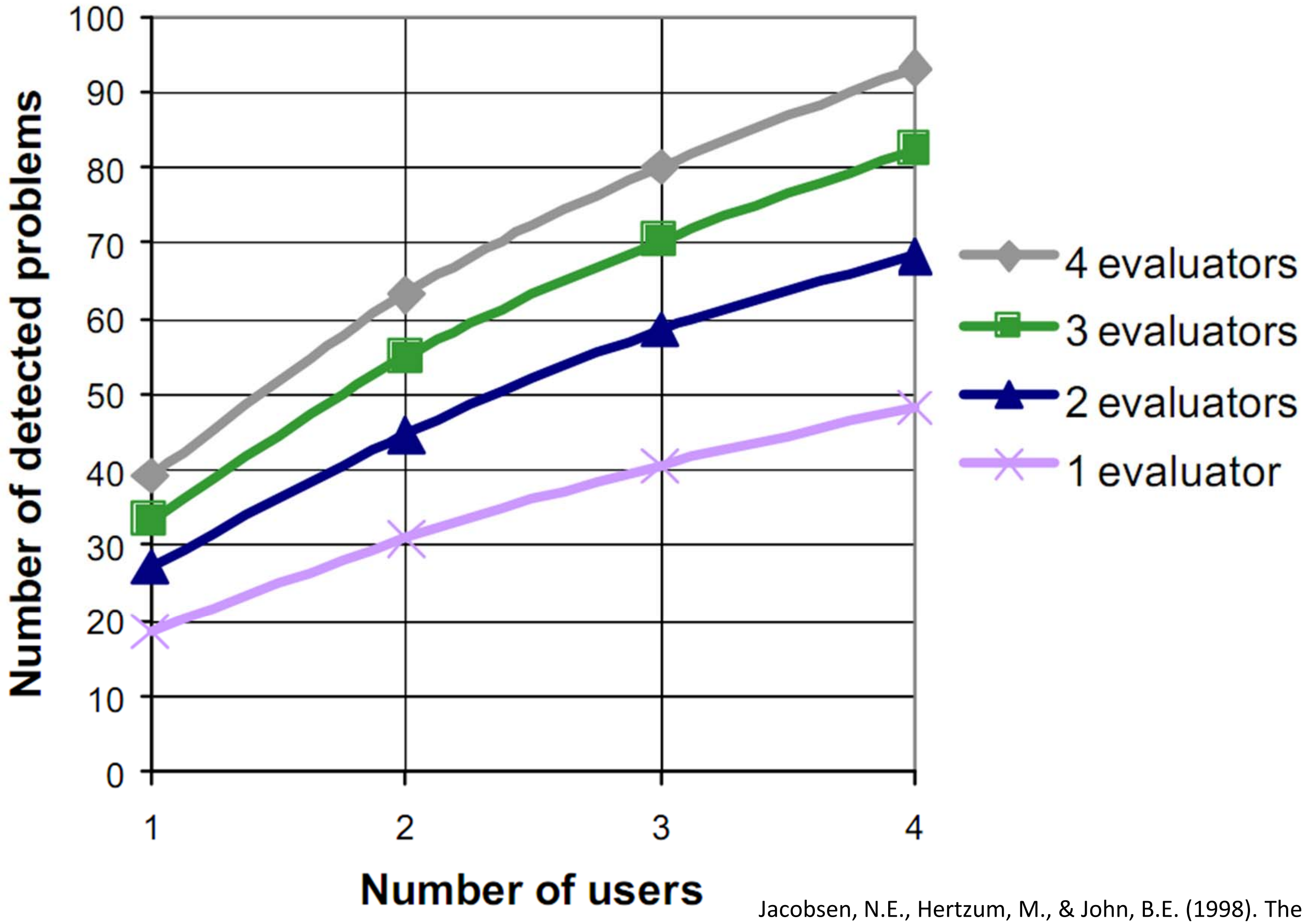
Mäntylä & Itkonen, Manuscript in preparation

Requirement review – addition of reviewers leads addition of detected defects in all processes (OPT/BIT) and effort combinations (2h, 4h, 6h)



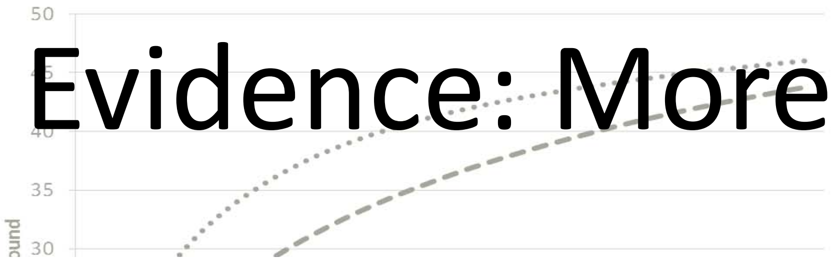
Biffle & Halling, " Investigating the defect detection effectiveness and cost benefit of nominal inspection teams" IEEE TSE 29(3), 2003

Usability: Observing think aloud users with different number of evaluators



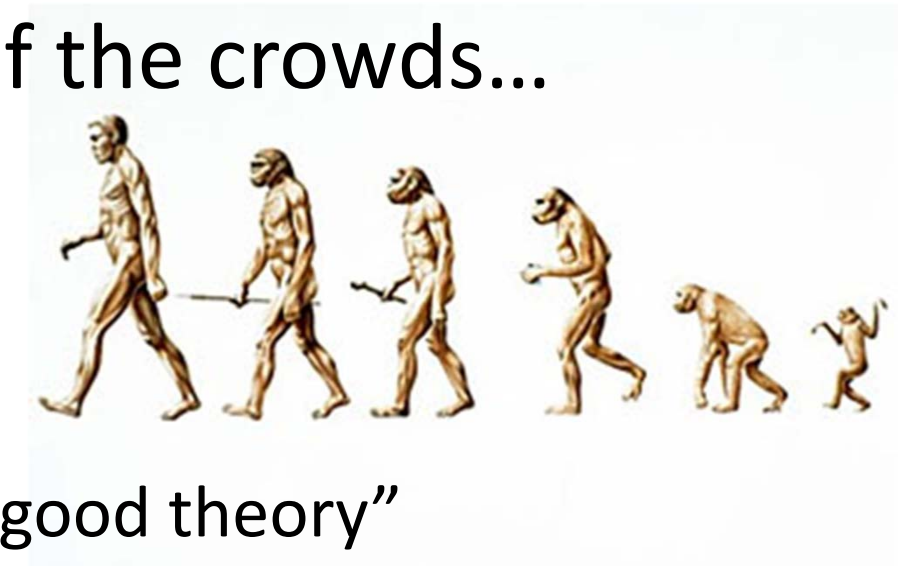
Jacobsen, N.E., Hertzum, M., & John, B.E. (1998). The evaluator effect in usability studies: Problem detection and severity judgments.

Evidence: More is better



- Support for
  - Theory of Software QA tasks being additive
    - In both QA and car pushing there is a ceiling effect
      - In QA due to max number of defects
      - In car pushing due to limited spots where can be pushed (see figure)
  - Linus law *given enough eyeballs, all bugs are shallow*
- The benefit of second opinion in Software QA (meta analysis of 5 research articles)
  - 1->2 individuals ~50% (1/2) unique defects
- Expertise and effort matters
  - But can be substituted with several individuals
    - Using less effort
    - Having lesser expertise

Outline - Power of the crowds…



- Theoretical background
  - “Nothing is as practical as a good theory”

- Examples and evidence

- **Time restricted crowds in testing**

- Downsides of crowds

Great, more is better but …
more testers -> more cost

Experiment: Crowds and Division Labor

- A testing task consisting two features of a text editor (performed by 3-5 year students)
  - Search and replace
  - Editing source code (Tabbing and Indentation, Bracket Matching, etc)
- Which one would you pick to find as many defects as possible?
  - Single tester using as much time as he needs (avg 9.83h)
    - Few eyes but more thorough
  - Multiple time-restricted (TR) testers using 2h each (non-communicating)
    - More eyes
- Many TR testers complained about the lack of time: 29% (in the open questions)
  - “Time run out”, “Short time. Not very realistic test.”, “Alue oli mielestäni liian laaja aikaan nähden jos meinattiin että kaikki kohdat testataan kunnolla.”, ”Molemmissa testaussessioissa aikarajoite tuntui hankalalta”, ” I think that the biggest problem is time. If I had more time, I should do an exploratory testing more deep.”
- When do time-restricted (TR) (2h) testers beat single non time restricted (NTR) tester (9.83h avg)?
  - 2 TR testers -> 4h vs. 9.83h 3 TR testers -> 6h vs. 9.83h
  - 4 TR testers -> 8h vs. 9.83h 5 TR testers -> 10h vs. 9.83h
  - 6<=TR testers -> 12++h vs. 9.83h NTR tester always better

Lets try it out!

- You have two tasks: first 200s, second 100s
- With pen mark as many defects as you can find
  - Please, mark the order in which you found the defects
  - Please ignore defects due to copy-machine

Defects

- All defects are easy to see once you know where to look

Wrong defects

- Please ignore defects due to copy machine and low quality microfilm
- If unsure mark it as defect

- You have two tasks: first 200s, second 100s
- Write your gender and age on the back side
- With pen mark as many defects as you can find
  - Please, mark the order in which you found the defects

```
(1)     (2)     (3)     (4)     (5)
    (6)     (7)     (8)     (9)   (10)
```

Average number of unique defects found by testers and 5%-95% range

Testers	TR testers (2h)	NTR testers (9.83h)	2*TR testers (2h)	3 * TR testers (2h)	4 *TR testers (2h)	5* TR testers (2h)
1	7,53 (4-11)	11,3 (4-18)				
2						
3						
4						
5						
6						
7						
8						
9						
10						

- Testers that have more time find more defects…

Average number of unique defects found by groups of testers and 5%-95% range

Testers	TR testers (2h)	NTR testers (9.83h)	2*TR testers (2h)	3 * TR testers (2h)	4 *TR testers (2h)	5* TR testers (2h)
1	7,53 (4-11)	11,3 (4-18)	11,98 (8-16)			
2						
3						
4						
5						
6						
7						
8						
9						
10						

- Using two testers that do not have enough time gives equal result in comparison to single tester with enough time
- Effort is saved 9.83h vs. 4h

Average number of unique defects found by groups of testers and 5%-95% range

Testers	TR testers (2h)	NTR testers (9.83h)	2*TR testers (2h)	3 * TR testers (2h)	4 *TR testers (2h)	5* TR testers (2h)
1	7,53 (4-11)	11,3 (4-18)	11,98 (8-16)	15,06 (11-20)	17,41 (13-22)	19,33 (15-24)
2						
3						
4						
5						
6						
7						
8						
9						
10						

- 3-5 time resticted testers beat one non time restricted tester

Average number of unique defects found by groups of testers and 5%-95% range

Testers	TR testers (2h)	NTR testers (9.83h)	2*TR testers (2h)	3 * TR testers (2h)	4 *TR testers (2h)	5* TR testers (2h)
1	7,53 (4-11)	11,3 (4-18)	11,98 (8-16)			
2	11,98 (8-16)	16,93 (11-23)	17,41 (13-22)			
3	15,06 (11-20)	20,56 (15-27)	20,94 (16-26)			
4	17,41 (13-22)	23,29 (18-29)	23,62 (19-29)			
5	19,33 (15-24)	25,52 (20-31)	25,80 (21-31)			
6	20,94 (16-26)	27,36 (22-33)	27,66 (23-33)			
7	22,35 (18-27)	28,95 (24-34)	29,25 (25-34)			
8	23,62 (19-29)	30,33 (25-36)	30,69 (26-36)			
9	24,76 (20-30)	31,55 (27-37)	31,96 (27-37)			
10	25,80 (21-31)	32,63 (28-38)	33,13 (28-38)			

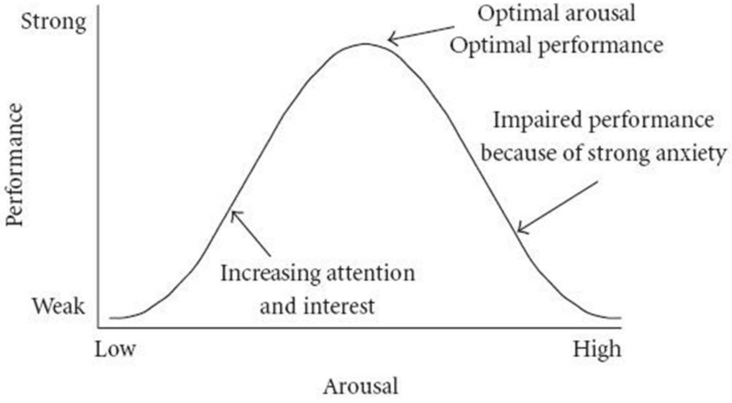
- Relationship 1NTR = 2TR holds when testers are added

Average number of unique defects found by groups of testers and 5%-95% range

Testers	TR testers (2h)	NTR testers (9.83h)	2*TR testers (2h)	3 * TR testers (2h)	4 *TR testers (2h)	5* TR testers (2h)
1	7,53 (4-11)	11,3 (4-18)	11,98 (8-16)	15,06 (11-20)	17,41 (13-22)	19,33 (15-24)
2	11,98 (8-16)	16,93 (11-23)	17,41 (13-22)	20,94 (16-26)	23,62 (19-29)	25,80 (21-31)
3	15,06 (11-20)	20,56 (15-27)	20,94 (16-26)	24,76 (20-30)	27,66 (23-33)	30,00 (25-35)
4	17,41 (13-22)	23,29 (18-29)	23,62 (19-29)	27,66 (23-33)	30,69 (26-36)	33,13 (28-38)
5	19,33 (15-24)	25,52 (20-31)	25,80 (21-31)	30,00 (25-35)	33,13 (28-38)	35,59 (31-40)
6	20,94 (16-26)	27,36 (22-33)	27,66 (23-33)	31,96 (27-37)	35,14 (31-40)	37,64 (33-42)
7	22,35 (18-27)	28,95 (24-34)	29,25 (25-34)	33,66 (29-38)	36,88 (32-41)	39,38 (35-43)
8	23,62 (19-29)	30,33 (25-36)	30,69 (26-36)	35,14 (31-40)	38,37 (34-43)	40,93 (37-45)
9	24,76 (20-30)	31,55 (27-37)	31,96 (27-37)	36,46 (32-41)	39,7 (35-44)	42,24 (38-46)
10	25,80 (21-31)	32,63 (28-38)	33,13 (28-38)	37,64 (33-42)	40,93 (37-45)	43,4 (40-46)

Some possible reasons

- Individuals find different defects
- Fresh eyes effect
  - How many hours of testing is needed before a tester is blind to the defects in the software under test
  - Early hours more effiecient than late hours
  - “*I have known men who could see through the motivations of others with the skill of a clairvoyant(=selvännäkijä); only to prove blind to their own mistakes. I have been one of those men*.”- Bernard M. Baruch
- Overspending
  - Working on your own is less efficient than working under control
  - NTR testers do not know when to stop
- Postive effects of the dead line
  - Yerkes–Dodson law



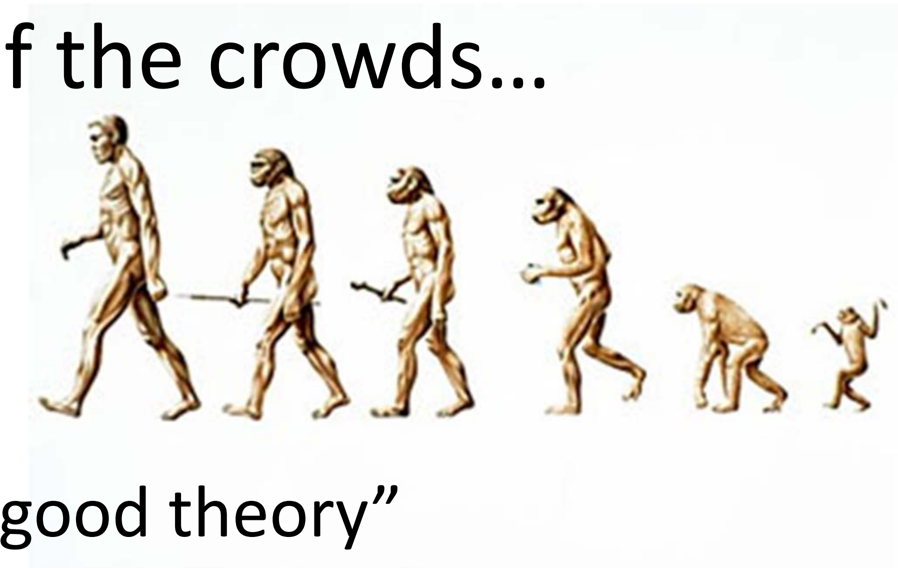
Practical implication: Divide testing tasks to have redundancy

	Tim	Tom	Tammy
TestTask 1	X		
TestTask 2		X	
TestTask 3			X
TestTask 4	X		
TestTask 5		X	
TestTask 6			X

	Tim	Tom	Tammy
TestTask 1	X	X	
TestTask 2		X	X
TestTask 3	X		X
TestTask 4	X	X	
TestTask 5		X	X
TestTask 6	X		X

Is more really better…

Outline - Power of the crowds…



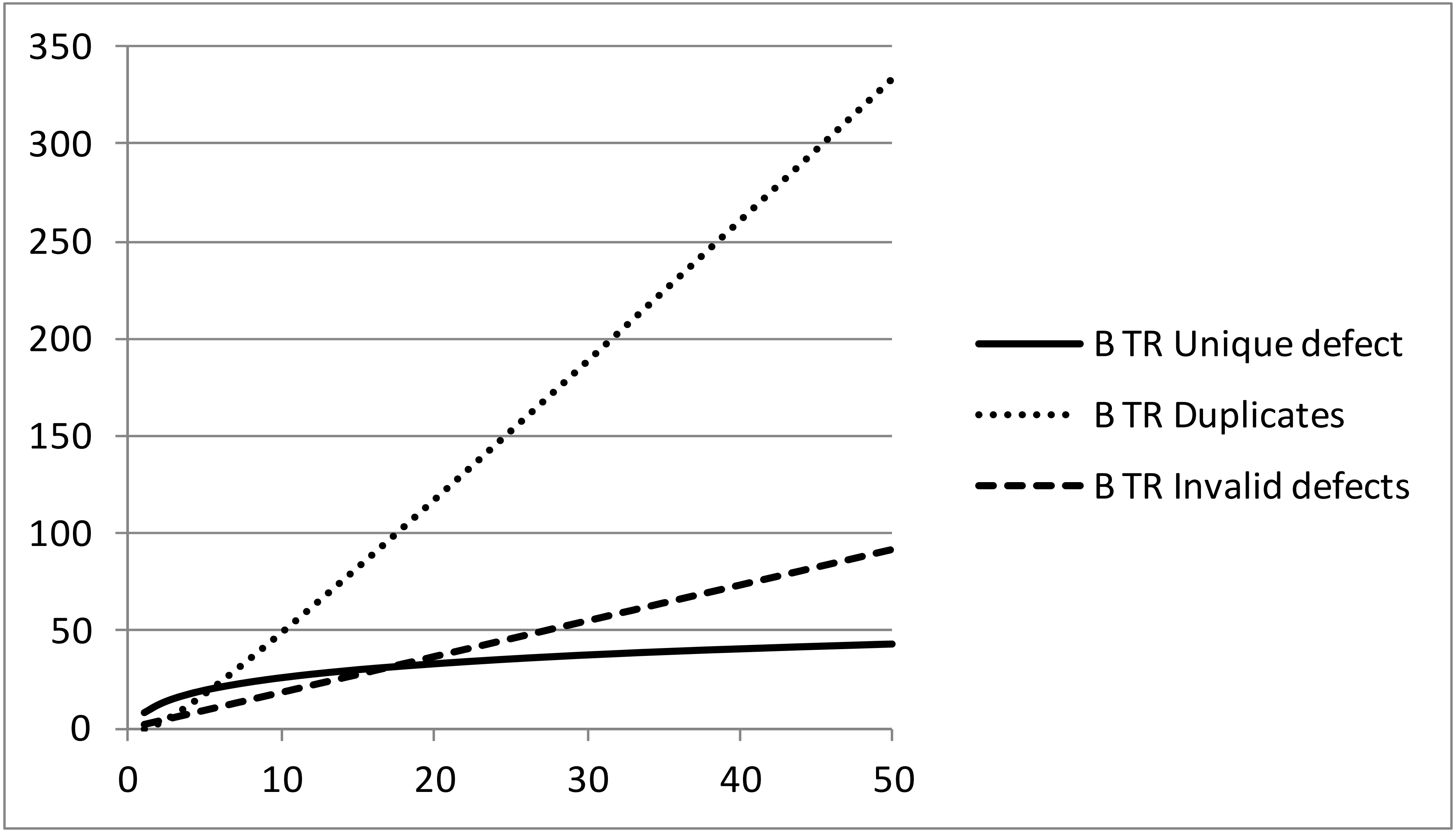
- Theoretical background
  - “Nothing is as practical as a good theory”

- Examples and evidence

- Time restricted crowds in testing

- **Downsides of crowds**

With many testers duplicate defects dominate

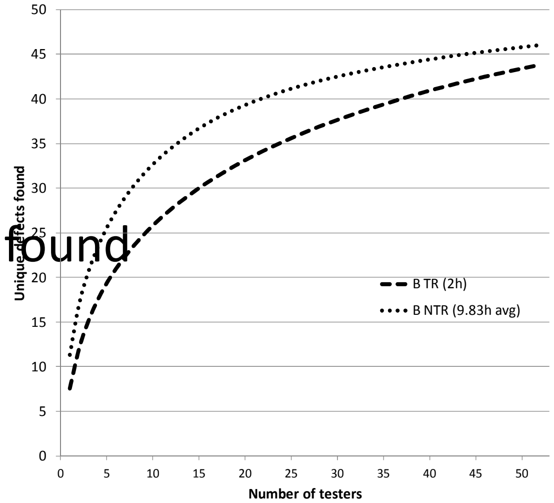


Downsides of multiple testers

- Duplicate defect reports
  - Evidence exists that duplicates can be useful!
    - Give extra information of defects (Bettenburg 2008)
  - Can be considered as indication failure frequency and used in bug prioritization
    - You want to fix more frequently occurring defects
- Invalid defect reports
  - Ones that are not defects after all or
  - Ones that are reported in incomprehensible way
- Need to combine results

Summary - Power of the crowds…

- Theoretical background - Steiner's taxonomy of tasks
  - Testing and QA is additive
    - With ceiling effect
  - Increase in expertise and effort helps, but still additive
- Examples and evidence
  - Testing crowds (internal usage, hire online, involve customers)
  - Conclusive evidence *more is better*
  - Benefits of 2nd tester: 50% (1/2) more unique defects found
- Time restricted crowds in testing and QA
  - 1*NTR tester (9.83h) = 2*TR testers (4h)
  - More evidence still needed
- Duplicate defect report handling is key when using multiple testers
  - Duplicates are not only negative
    - Provide extra info developers,
    - Defect occurrence frequency (+1)


Unique defects found
Number of testers
B TR (2h)
B NTR (9.83h avg)