

# First Results from an Investigation into the Validity of Developer Reputation

(Christian R. Prause and Markus Eisenhauer)

## Derived from Wiki Articles and Source Code



Developers thinking about others while they write collaboratively

Reputation forming

good

ok

good

poor

ok

Reputation computing

Do ranks match?

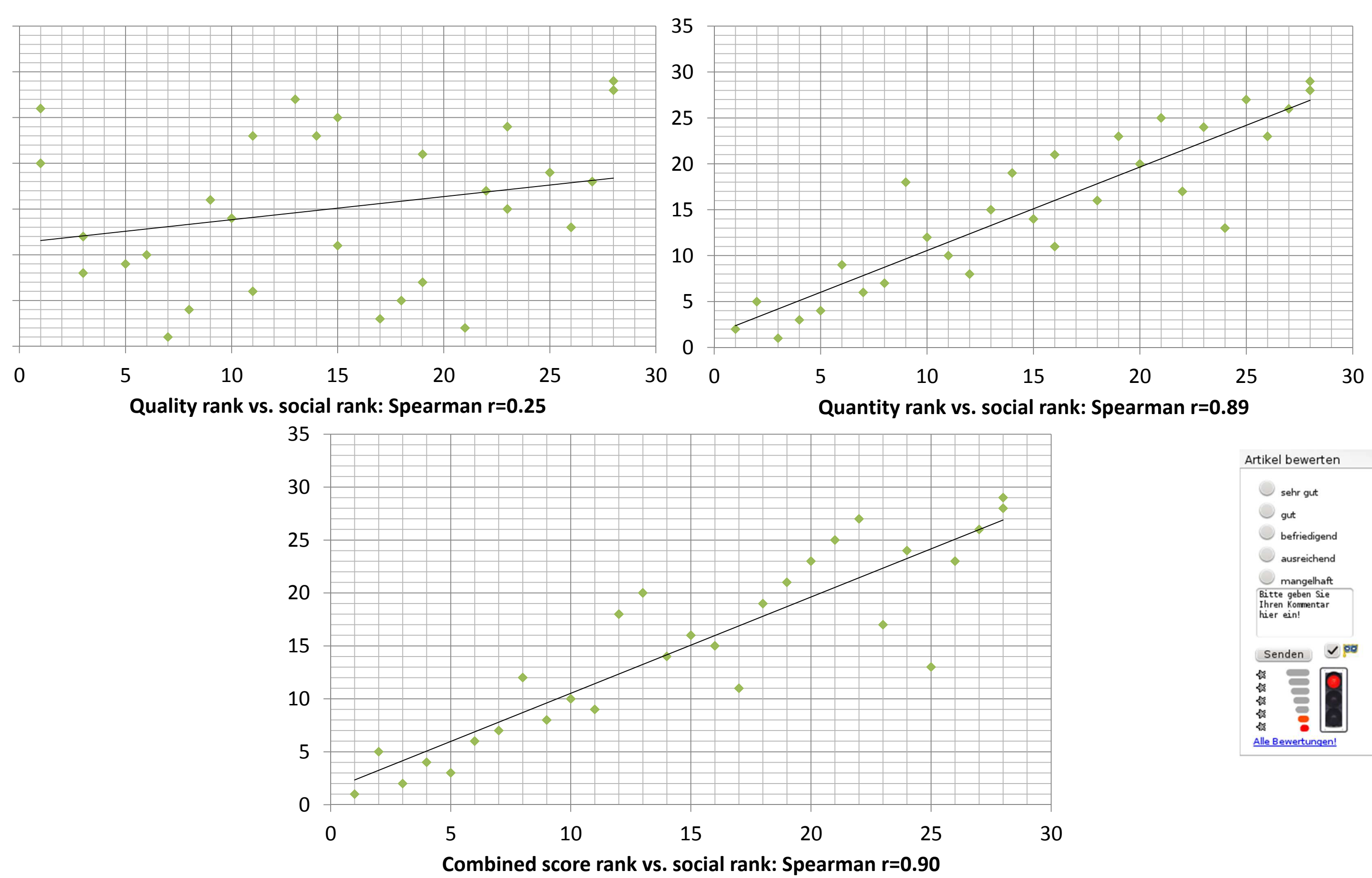
Social

Computed

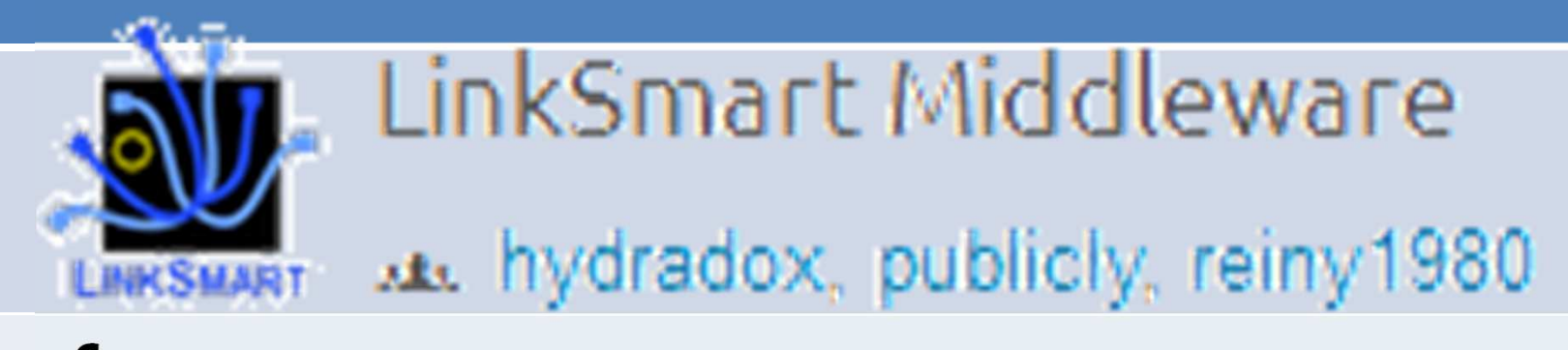
### Moknowpedia Wiki



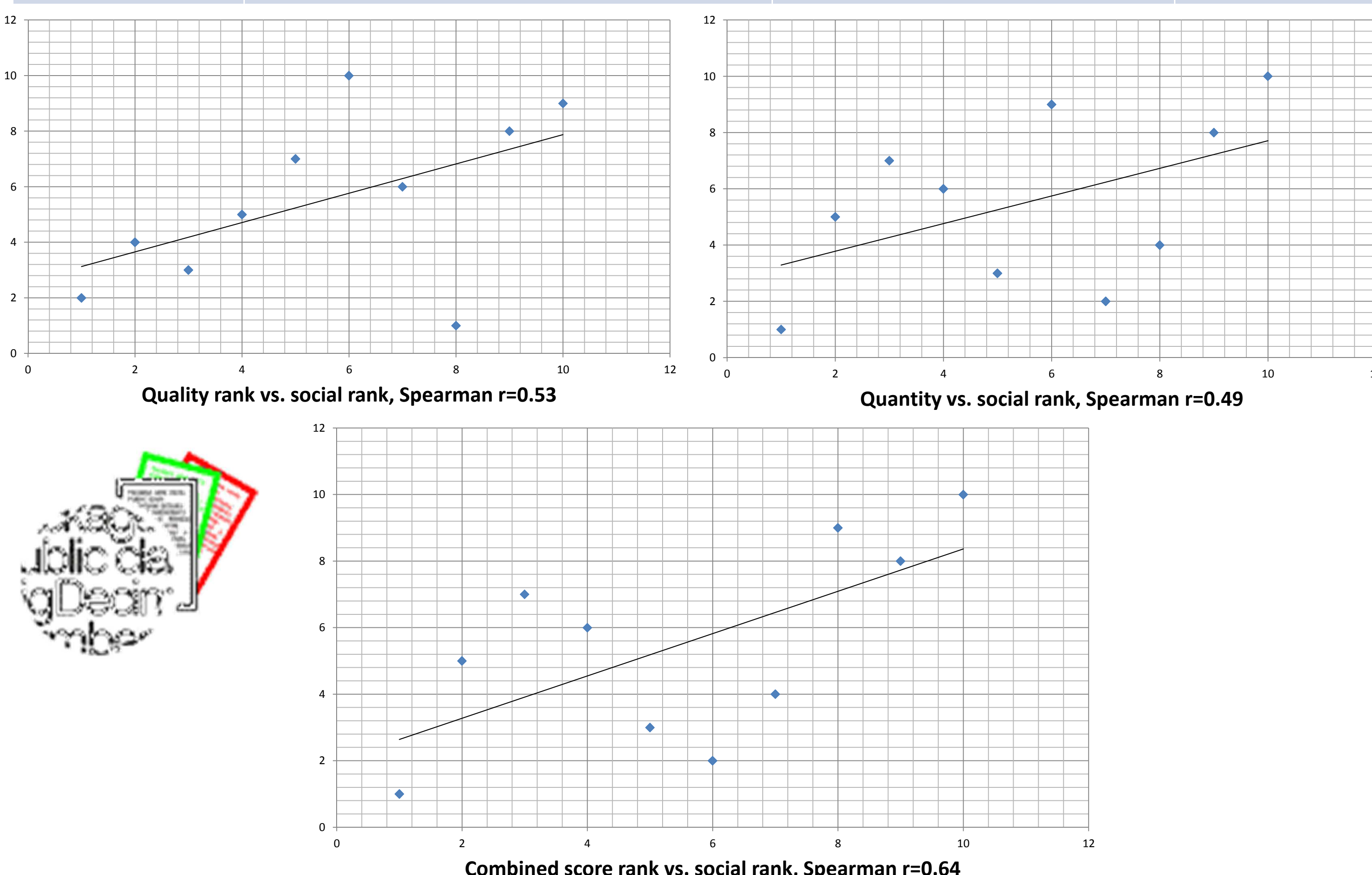
Type:	Internal knowledge management		
Text:	Wiki articles	Artifacts:	7000
Ratings:	Manual	# of ratings:	800
Editors:	50	Interviewed:	14



### LinkSmart Middleware



Type:	Open source software		
Text:	Java source code	Artifacts:	5000
Ratings:	Checkstyle	# of ratings:	1200
Editors:	10	Interviewed:	8



*“Developers don’t like to do documentation, because it has no value for them”*

**Idea:** Reputation system to improve internal quality of software by motivating developers to write documentation, i.e. wiki articles or clean source code with comments.

**Concept:** Someone who has made many and good contributions should have a high reputation in the community. Having contributed less or having contributed worse quality should lead to a lower reputation score.

**How:** Compute reputation score from artifact size, authorship and quality ratings. Authorship from evolution history: who authored respective text passage? Quality ratings from manual or automatic reviews.

**Research:** Is computed “documentation reputation” rank equal to social reputation rank? Interviews to compare social rank with computed quality, quantity and combined contribution rank.

**Result:** Computed reputation scores are valid. High correlations for quantity and combined rank, for source code also for quality. Reputation systems can work. Awareness is important for reputation.