	effect	effect's goal level influenced by	promoted and inhibited process variants
desired	better code quality	+ long-term thinking + inexperienced/heterogeneous developers + frequent need to change old code + code standards and rules as check baseline + complex or large codebase - separation of developer responsibilities (e.g. one developer per module) - "features are everything" culture	+ selection of reviewers with a grasp for architecture and the larger context + selection of reviewers with knowledge of the used libraries (to spot duplicate code)
	finding defects	+ defects have severe consequences (high costs, high service effort, feature will actually be used at all, ...) + no tests/testing is difficult + inexperienced developers - intensive testing, high trust in tests - defects that are hard for humans to find - no skilled reviewers available - no accountability for quality	+ comparison with a reference (e.g. requirements document) + selection of reviewers with high knowledge of the reviewed module + understanding of the changes (not just skimming) + checking of tests + following up on issues, re-review - permanent, face-to-face interaction
	learning of the reviewer	+ large differences in knowledge levels + collective code ownership + complex or large codebase - separation of developer responsibilities - small, well-rehearsed team - alternative knowledge transfer measures	+ permanent, face-to-face interaction + selection of reviewers missing knowledge of the reviewed module - fixed reviewer cadre
	learning of the author	+ large differences in knowledge levels + collective code ownership + complex or large codebase - separation of developer responsibilities - small, well-rehearsed team - alternative knowledge transfer measures	+ permanent, face-to-face interaction + selection of reviewers with more/different knowledge than the author + positive remarks + early feedback + reviews (even) for exercises or tutorials - "fixing on the fly" by the reviewer
	sense of mutual responsibility	- separation of developer responsibilities	+ permanent, face-to-face interaction + positive remarks
	finding better solutions	+ no discussion of designs before implementation + doubts regarding a specific solution (individual level)	+ selection of reviewers with different experiences and points of view + discussion in reviews + permanent, face-to-face interaction + early review (patch/pre commit) + parallel review - checklists
	complying to QA guidelines	+ certain industrial sectors + contractor relationship	+ like stated in the guidelines + documentation of reviews (e.g. by documenting issues)
undesired	staff effort	+ review effort not considered in plan + high work pressure + feature-oriented culture + consultant/contractor relationship	+ distribution of review effort on larger reviewer population + pull-based assignment + varying intensity of checks + early feedback - higher number of reviewers - face-to-face interaction (meetings) - very small reviews (too much overhead)

effect		effect's goal level influenced by	promoted and inhibited process variants
undesired	increased cycle time	+ time is short (e.g. due to very short sprints) + review is scheduled near a deadline + review duration not considered in plan	+ process measures to reduce waiting reviews (WIP limit, prioritization, ...) + fixed integration into the development process + selective rework - face-to-face interaction - higher number of reviewers - sequential review - very large reviews (other tasks will be stuck in the meantime)
	offending the author	+ error culture: errors are seen as personal failures + missing ability to receive criticism among the potential authors + missing ability to give criticism among the potential reviewers + separation of developer responsibilities + developers with a monopoly on knowledge + unskilled developers (leading to a high number of issues) - collective code ownership	+ face-to-face discussion of issues + small number of reviewers + objective and fair rules for review (e.g. reviews for every change) + private communication of issues + taking care when phrasing issues
	varying results	+ varying attitudes towards quality	+ checklists + higher number of reviewers + processes depending less on human judgment

Note: A plus sign in front of an influencing factor means "makes this effect more important", a minus sign means "makes this effect less important". A plus sign in front of a process variant means "is promoted when this effect is more important", a minus sign means "is inhibited when this effect is more important".
This table belongs to the research article "Factors Influencing Code Review Processes in Industry" by Tobias Baum, Olga Liskin, Kai Niklas and Kurt Schneider (Leibniz Universität Hannover).