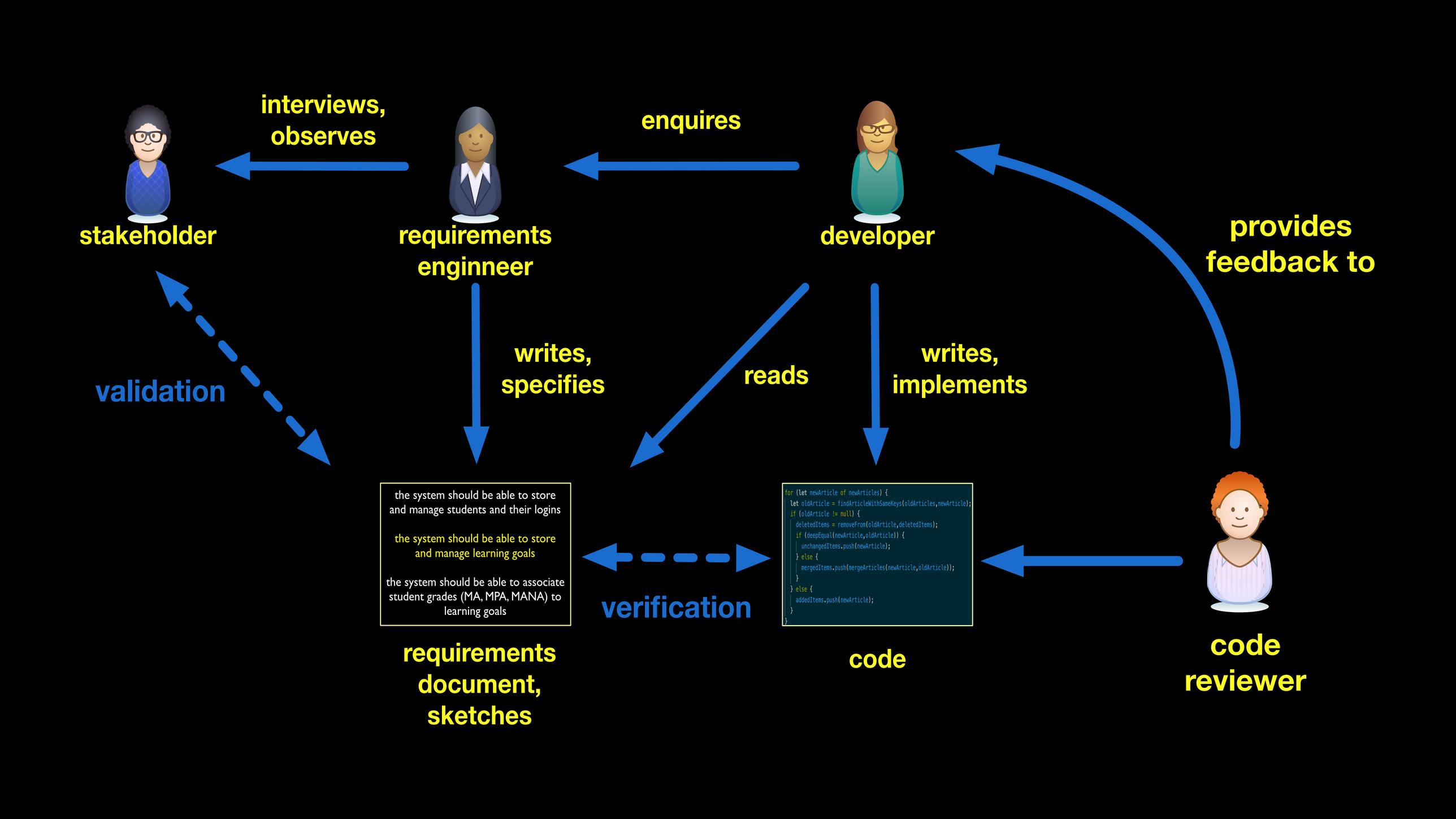
Software Engineering

Paulo Borba
Informatics Center
Federal University of Pernambuco

pauloborba.cin.ufpe.br

Code review


interviews,
observes
enquires
stakeholder
requirements
enginneer
developer
provides
feedback to
validation
writes,
specifies
reads
writes,
implements
the system should be able to store and manage students and their logins
the system should be able to store and manage learning goals
the system should be able to associate student grades (MA, MPA, MANA) to learning goals
verification
for (let newArticle of newArticles) {
let oldArticle = findArticleWithSameKeys(oldArticles,newArticle);
if (oldArticle != null) {
deletedItems = removeFrom(oldArticle,deletedItems);
if (deepEqual(newArticle,oldArticle)) {
unchangedItems.push(newArticle);
} else {
mergedItems.push(mergeArticles(newArticle,oldArticle));
}
} else {
addedItems.push(newArticle);
}
}
requirements
document,
sketches
code
code
reviewer

Code review snippets

maybe rename to getTypeName?

extract as dropFirstLine?

avoid duplication here

better to handle exceptions that will
be raised if diffj output format is changed

shoulnd’t this be a parameter?

please add provides and requires
interface for each service

if you are assuming that, it's safer to add
an assert stating it

the regular expression should be slightly more
complicated see reference manual, pg 23

Code review aspects

focus on different needs

- fixing, improving external quality factors (correctness, robustness, performance, security, etc.)
- refactoring, improving internal quality factors (readability, modularity, reusability, etc.)
- adding or refining documentation
- assuring coding styles and practices are adopted

Goes beyond verification, and quality assurance and conformance in general

Testing

Static Analysis

Code review

Review code and make notes

the addFilesToDirectory function

https://is.gd/34Zqz3

Code review report

the addFilesToDirectory function

- Large function, many functions could be extracted, revealing intention and improving readability
- Small code duplication, lines 54-62
- Directory structure is hardcoded, lines 104-115
- 137-151 could be a loop based on a map previously created
- No exception handling for file operations, lines162-168
- Poor documentation
- Legibility is OK
- Variable names are OK

How is code review actually done in practice?

Code reviewing in the trenches

Laura MacLeod et al. IEEE Software 2018.

- Four teams at Microsoft
- Ethnography-style observations (1 week)
- Semistructured contextual (firehouse) interviews with 18 developers
- Survey of 911 developers to validate initial findings
    - best practices
    - motivation
    - challenges

Code review is important

Reviewer feedback is useful

Review pressure makes developers more attentive

Increases confidence

Code review life cycle

not all teams have explicit rules or policies

- Preparation
- Selection of reviewers
- Notification of selected reviewers and stakeholders
- Feedback
- Iteration
- Check-in

~90% review code at least once a week

~40% review daily

most teams require code review before integration

Motivations for code review

1. Improve internal quality factors
2. Improve external quality factors (find defects)
3. Transfer knowledge
4. Explore alternative solutions
5. Improve the development process
6. Avoid breaking builds
7. Increase team awareness and integration
8. Share code ownership
9. Team assessment

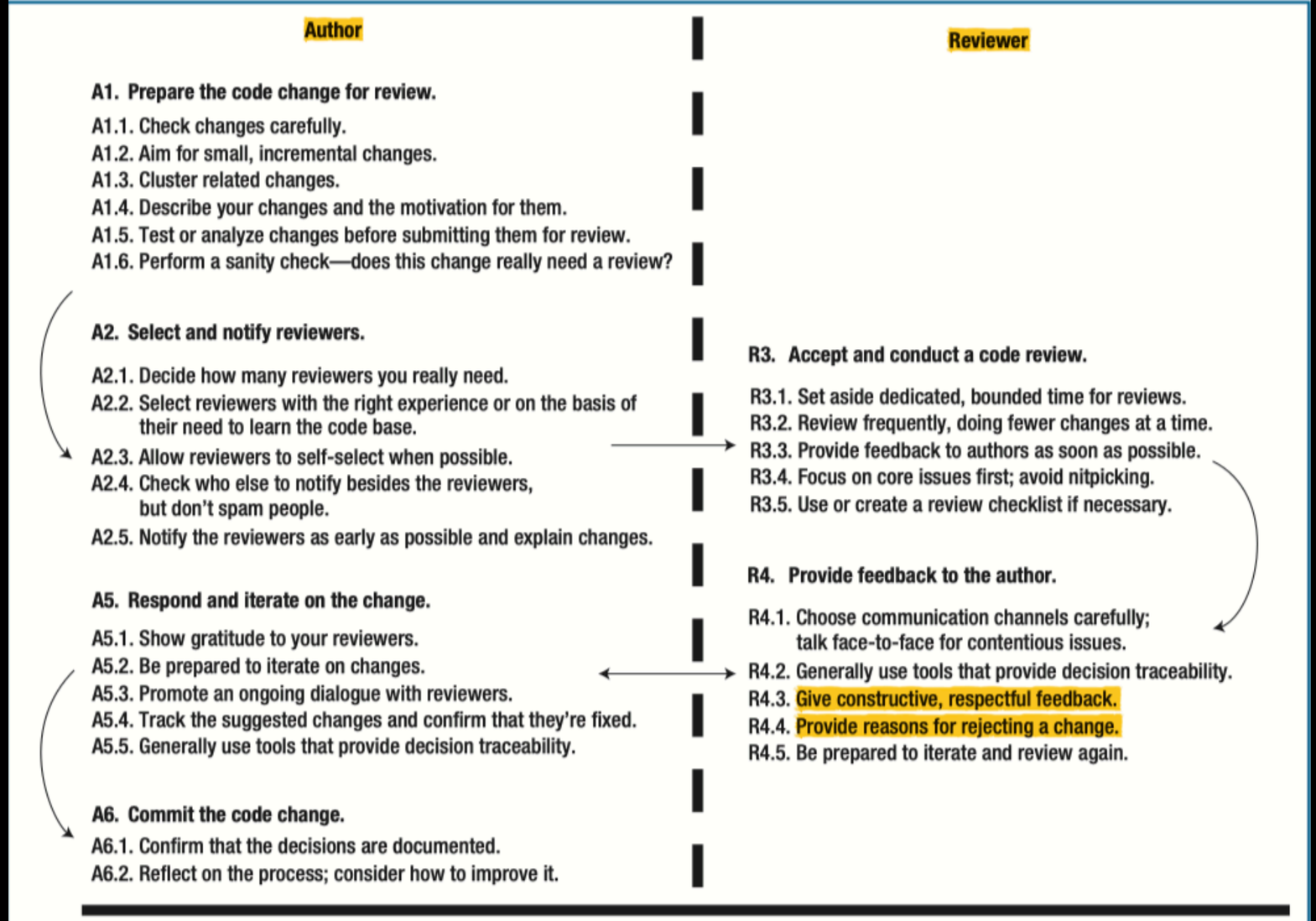
Code review challenges for the author

1. Receiving timely feedback
2. Receiving insightful feedback
3. Finding reviewers
4. Documenting changes to be reviewed
5. Managing multiple communication channels (code review system, chat, face-to-face)
6. Dealing with rejections
7. Reaching consensus (bikeshedding)

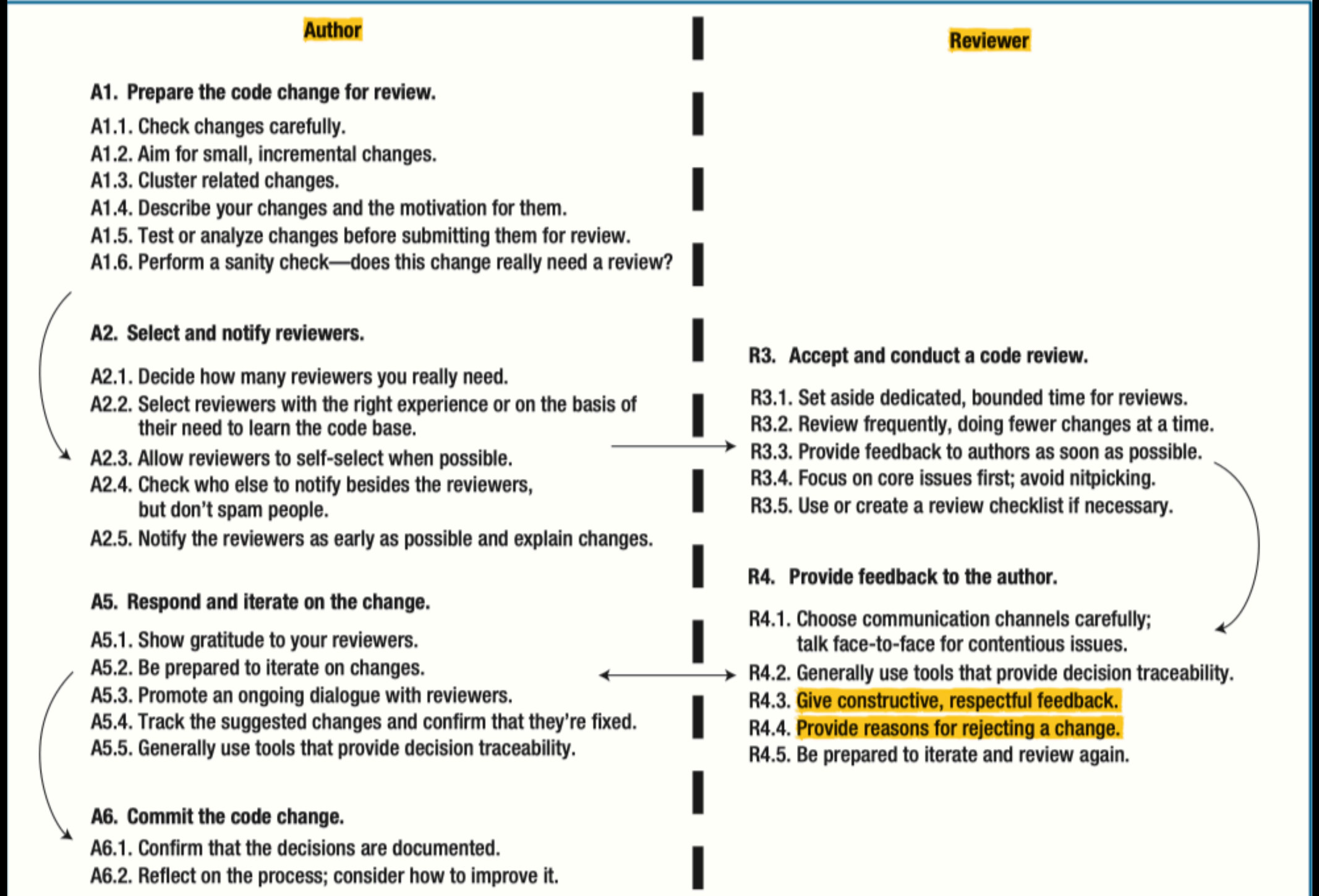
Code review challenges for the reviewer

1. Large (tangled?) changeset
2. Finding time for properly reviewing code
3. Understanding the motivation for the change
4. Understanding how the change was implemented
5. Understanding change history and decisions
6. Insufficient training
7. Reaching consensus (bikeshedding)

Code review practices



Code reviewing in the trenches.
Laura MacLeod et al.
IEEE Software 2018.


Author
Reviewer
A1. Prepare the code change for review.
A1.1. Check changes carefully.
A1.2. Aim for small, incremental changes.
A1.3. Cluster related changes.
A1.4. Describe your changes and the motivation for them.
A1.5. Test or analyze changes before submitting them for review.
A1.6. Perform a sanity check—does this change really need a review?
A2. Select and notify reviewers.
A2.1. Decide how many reviewers you really need.
A2.2. Select reviewers with the right experience or on the basis of their need to learn the code base.
A2.3. Allow reviewers to self-select when possible.
A2.4. Check who else to notify besides the reviewers, but don't spam people.
A2.5. Notify the reviewers as early as possible and explain changes.
R3. Accept and conduct a code review.
R3.1. Set aside dedicated, bounded time for reviews.
R3.2. Review frequently, doing fewer changes at a time.
R3.3. Provide feedback to authors as soon as possible.
R3.4. Focus on core issues first; avoid nitpicking.
R3.5. Use or create a review checklist if necessary.
R4. Provide feedback to the author.
R4.1. Choose communication channels carefully; talk face-to-face for contentious issues.
R4.2. Generally use tools that provide decision traceability.
R4.3. Give constructive, respectful feedback.
R4.4. Provide reasons for rejecting a change.
R4.5. Be prepared to iterate and review again.
A5. Respond and iterate on the change.
A5.1. Show gratitude to your reviewers.
A5.2. Be prepared to iterate on changes.
A5.3. Promote an ongoing dialogue with reviewers.
A5.4. Track the suggested changes and confirm that they're fixed.
A5.5. Generally use tools that provide decision traceability.
A6. Commit the code change.
A6.1. Confirm that the decisions are documented.
A6.2. Reflect on the process; consider how to improve it.

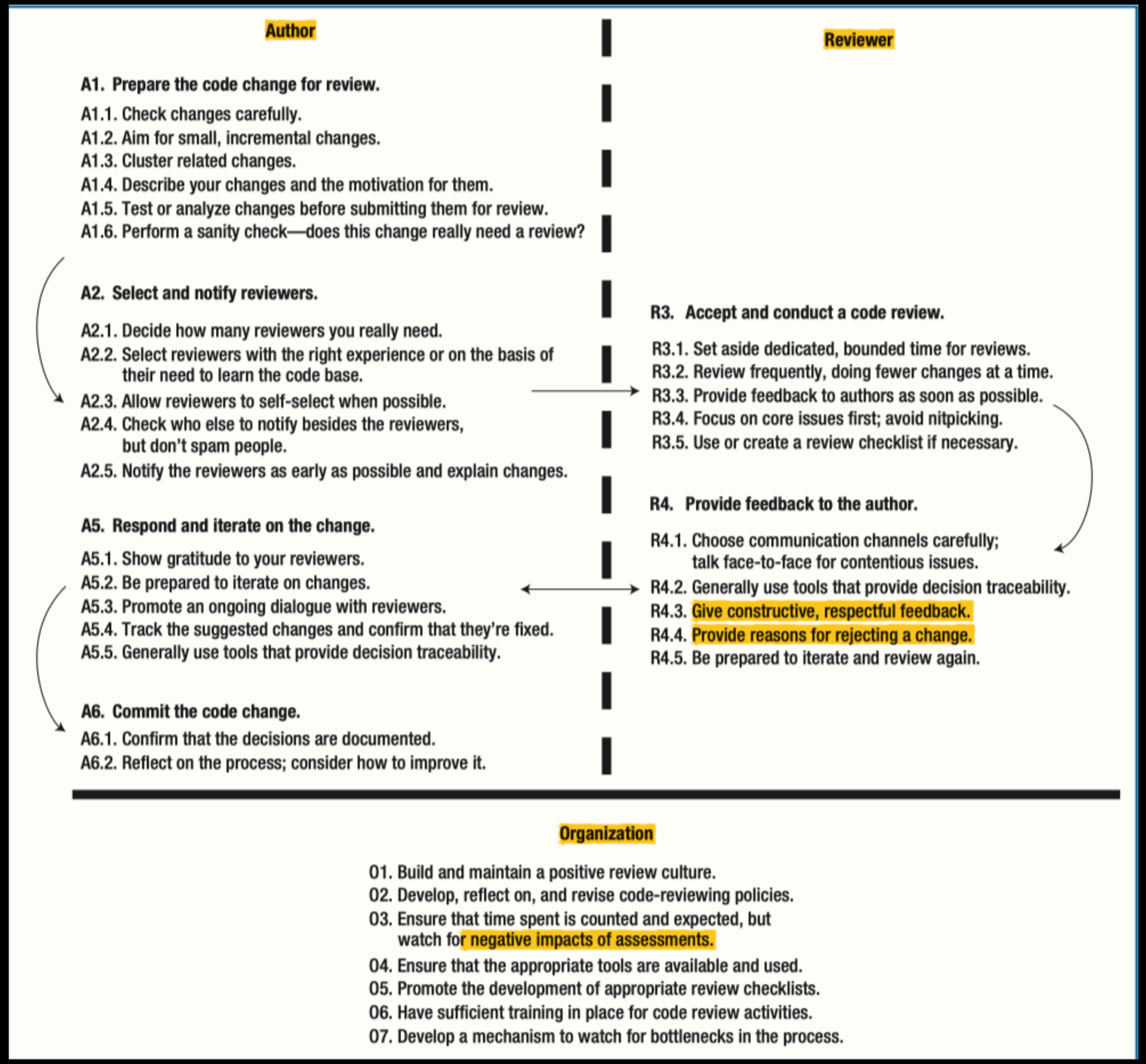
Organization

01. Build and maintain a positive review culture.
02. Develop, reflect on, and revise code-reviewing policies.
03. Ensure that time spent is counted and expected, but watch for negative impacts of assessments.
04. Ensure that the appropriate tools are available and used.
05. Promote the development of appropriate review checklists.
06. Have sufficient training in place for code review activities.
07. Develop a mechanism to watch for bottlenecks in the process.

Code reviewing in the trenches.
Laura MacLeod et al.
IEEE Software 2018.

Select at least one practice that

- you couldn't understand
- you think is relevant
- you think is irrelevant
- you have never practiced



Consider trade-offs when selecting practices

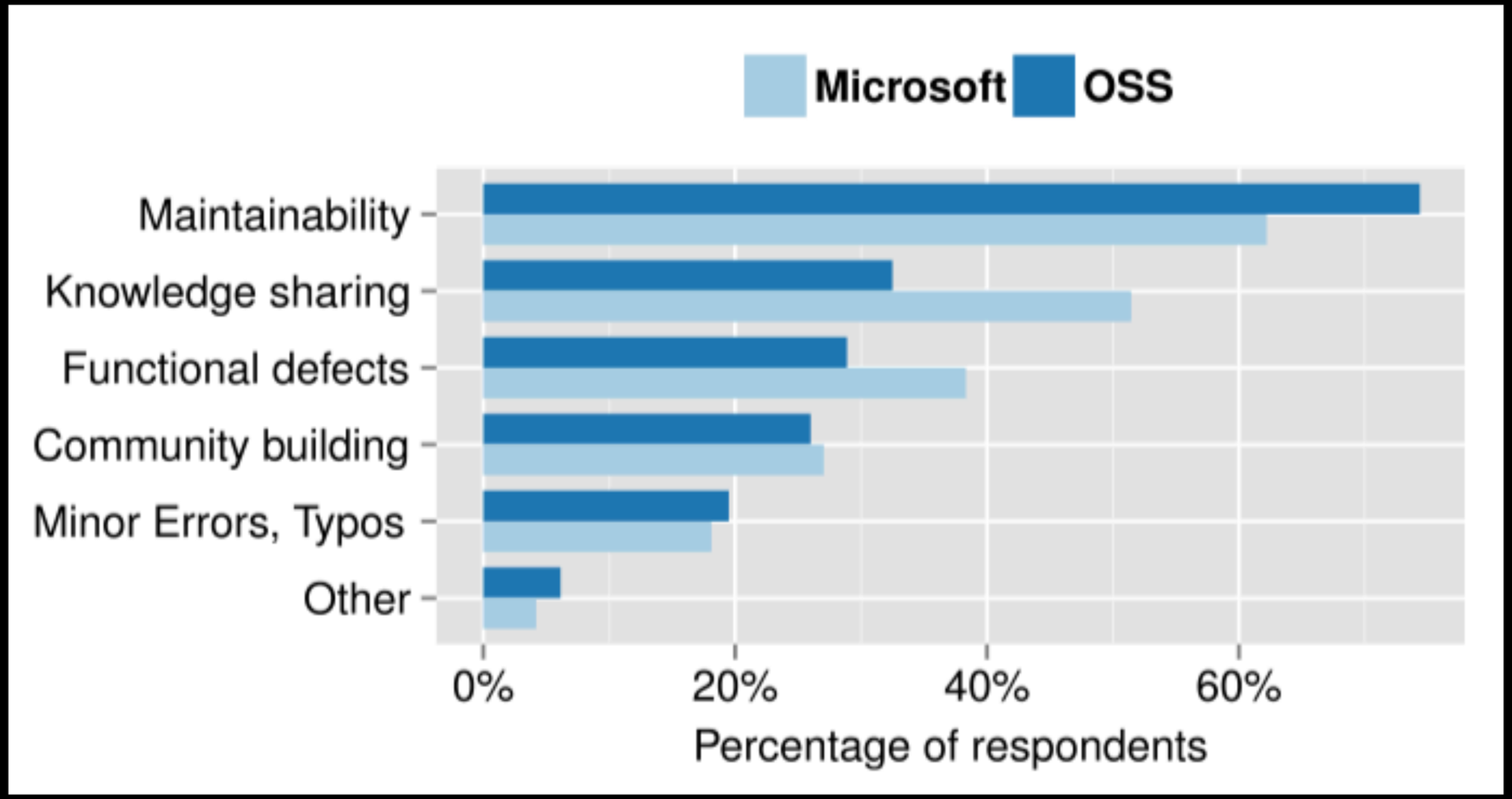
No code reviews for small changes that don’t affect code’s logic

Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft

Amiangshu Bosu et al. IEEE TSE 2017.

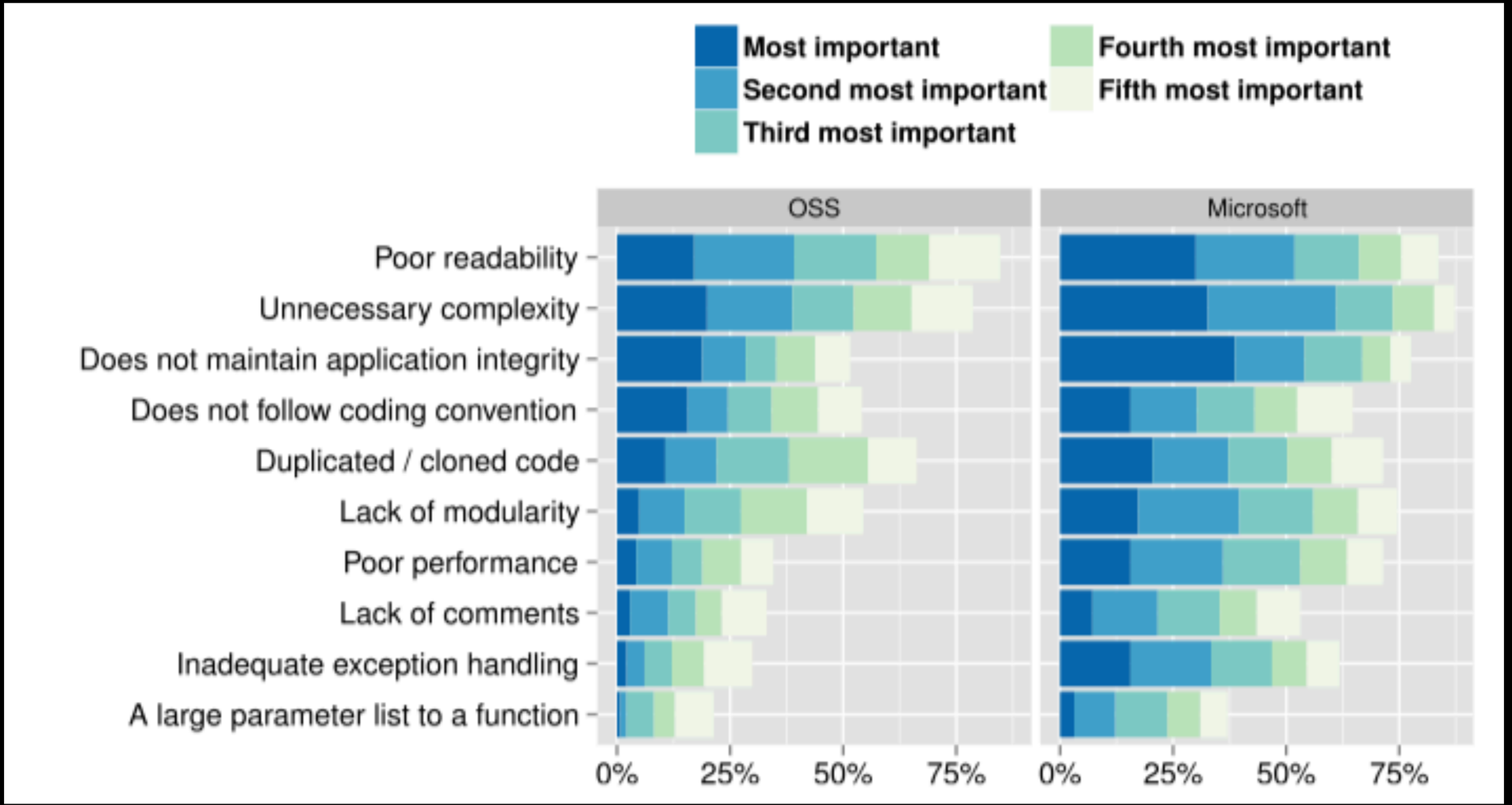
- Surveys of Microsoft and OSS developers
- Developers spend ~10-15% of their time with code reviews
  - more experienced developers spend more time
- Quality of the submitted code helps to form impression about authors
- Large similarity between OSS and Microsoft developers
  - main motivation in OSS is impression formation, contrasting with knowledge transfer at Microsoft

Code review motivation



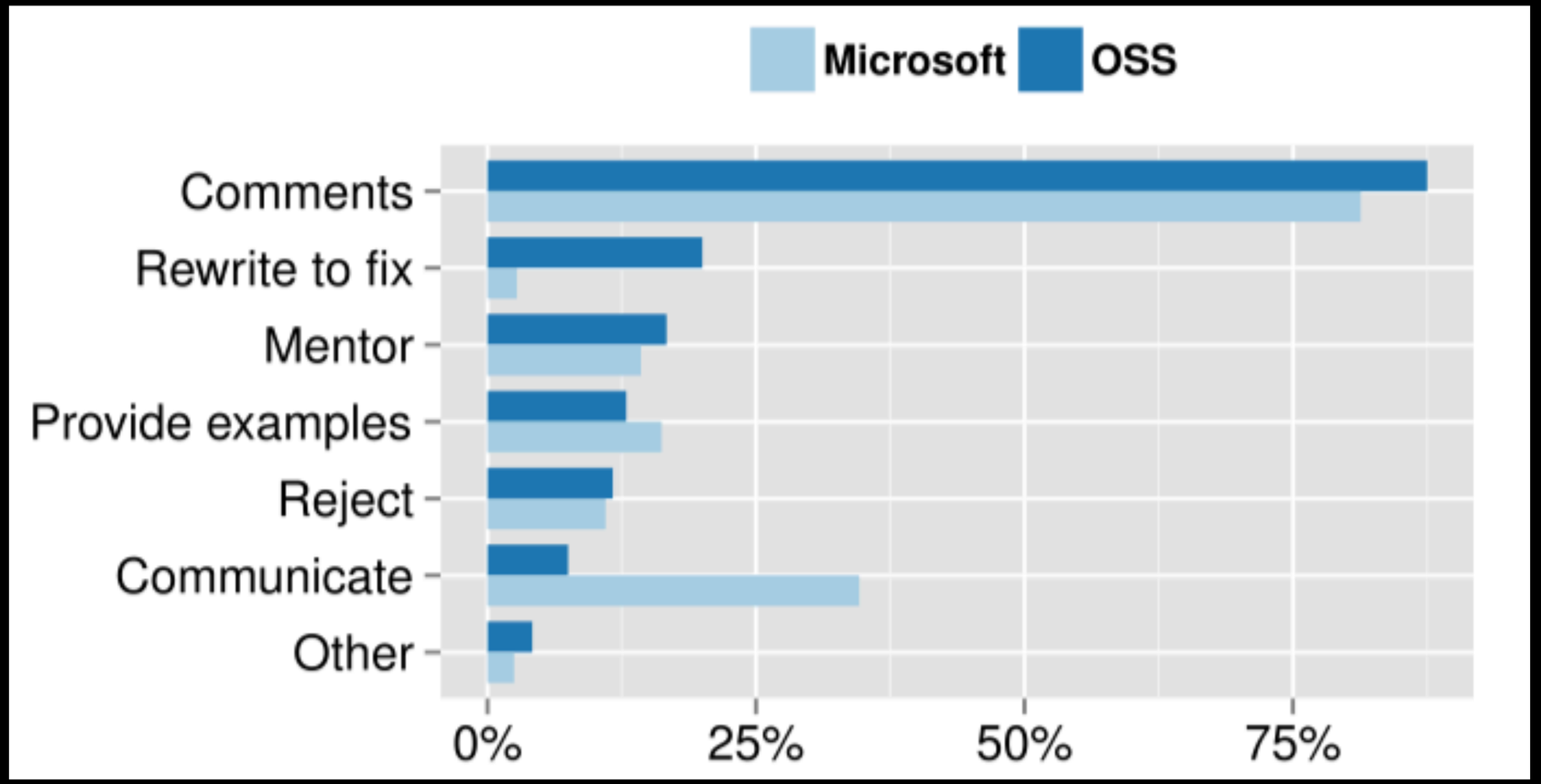
Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft. Amiangshu Bosu et al. IEEE TSE 2017.

Common detected characteristics of poor code



Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft. Amiangshu Bosu et al. IEEE TSE 2017.

How reviewers assist to fix poor code



Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft. Amiangshu Bosu et al. IEEE TSE 2017.

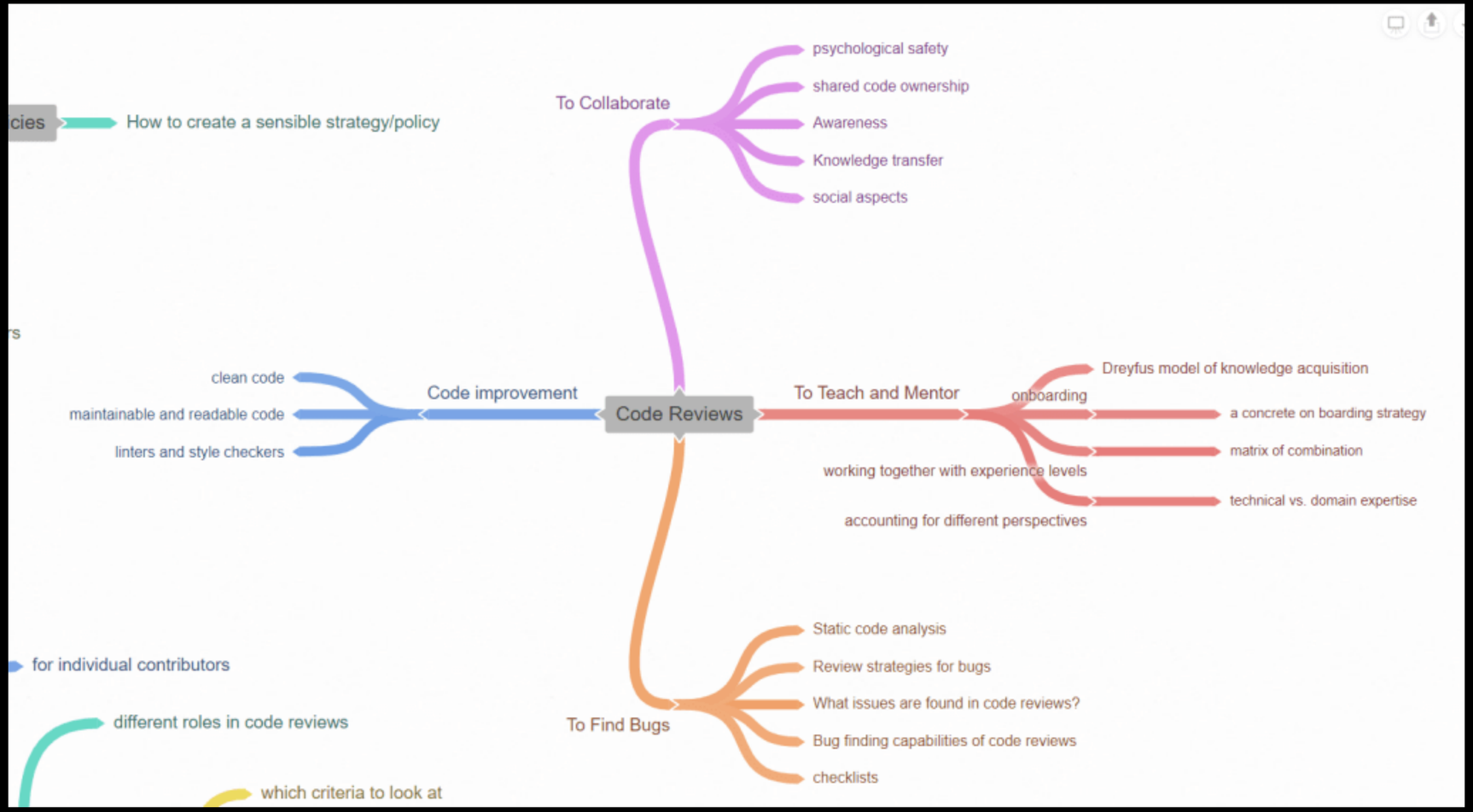
Code reviews
(lightweight, informal, tool-based)

versus

Inspections
(heavyweight, formal, meeting-based)

Code Review book and posts

Michaela Greiler



https://www.michaelagreiler.com/code-review-book/

Software Engineering

Paulo Borba
Informatics Center
Federal University of Pernambuco

pauloborba.cin.ufpe.br