



INTRODUCTION

- Wording and communication
- Periphery
- Breaks
- Test
- Source code needed for practical review :-)
- Server: <u>http://it-review.dhbw-stuttgart.de/</u>

MYSELF

- Johannes Unterstein, <u>unterstein@me.com</u>
- Finished DHBW in 2010
- Currently software developer at 1&1 Internet AG
 - huge and distributed eBusiness web application
 - e.g. eShops, WebAnalytics, Search engine marketing

WHO IS HERE?

- Which companies? Are you developers?
- Who worked in a team yet?
 - How did you handle this situation?
- Did your mentors reviewed your code?
- Did you participate a code review?
- What do you expect of this event?

WHY THIS EVENT?

- My first commit at 1&1
- Objective view of source code
 - Separation of code remark and personal criticism
- Better software quality
- Desired presence in the academic world



WHAT IS A REVIEW

- Review in general
 - Look at things, re-view
 - After factoring process
 - Reflect things
 - Check correctness
 - View with mental distance

TARGETS

- Code
- Components
- Architectures, Designs
- Refactorings, Tests
- Projects

PRECONDITIONS

• Team

- Ability of constructive criticism
- Common code base
- Common quality entitlement

- Administrative
 - Objective and diplomatic moderation
 - Management
 - Suitable software

CLASSIFICATION

- Important part of the software development process
- Tool to improve software quality
- No compensation for tests
- Continuous vs. dedicated
- It's not for free

WHY Continuous quality assurance Prevent bugs Distribution of knowledge ,,Acceptance ritual", new co-workers Estimate coding skills of team members Improve coding skills of all participants

WHAT KIND OF

- Classic code reviews
- Continuous code reviews
- Extreme programming / agile processes
- Open source

HOW TO USE

- Classic review process and tools
- Web based
- Commit mails (minimum)
 - Early recognition
- Mentor model
- Outsourcing

WHAT TO REVIEW

- Defined amount of code
 - Semantically closed piece of software
 - Single commits
 - Diff of two revision
- Size matters
 - Developers are expensive ;-)

WHO

- Roles
 - Author(s)
 - Organization
 - Reviewers
 - Moderator

- People
 - According team
 - As much as possible
 - Good mix



PREPARATION

- Initialization
- Organized by assistance or author(s)
- Which code?
- Which participants?
- Meeting
- Infrastructure

DISTRIBUTION

- Code
- Defined revision or tag
- Tools, accounts, access ...

INDIVIDUAL PHASE

- Each participant reviews the code for himself
- Makes remarks
- Discipline
- First step of knowledge transfer
- Be thorough and handle with care
- Take your time

THE REVIEW

- Plan enough time
- Don`t bother about usual discussions
- May not fit all organizations

- Process
 - All points sequentially
 - Decide if valid
 - Decide how to handle
 - Decide who will handle

REWORK AND CHECK

- Solve issues
- Mark tickets as resolved
- Check the rework and close the ticket



PROFIT - PEOPLE

- Author(s)
 - Receives feedback
 - Profits from the know how of other participants
- Reviewers
 - Know how exchange
 - Profits from the know how of other participants

PROFIT - CODE

- Prevent, reduces bugs
- Improves code quality
- Improves code, product, project knowledge
- Spreads knowledge

PSYCHOLOGIC ASPECTS

- Acceptance
 - Oh god, my code is reviewed!
 - Inspection, disgrace, transparency ...
- Objective view vs. take to be personal
 - The code isn`t you!
- Positive experience for the author



OUTSOURCING

- Pros:
 - Objectivity
 - Know how
 - New ideas

- Cons:
 - Acceptance
 - Know how
 - Internals
 - Best practices
 - Standards, ...