CODE REVIEWS DONE RIGHT

Heiko Gramlich

1&1

Agenda

- Über mich
- Code Review
- Code Review bei De-Mail Development
- Tooling
- Demo
- Erfahrungen im Team

Über mich

- Software Entwickler
- Werdegang:
  - Dipl. Informatiker der Medizin (2012)
  - 1&1 Source Center (Okt 2012 – Jan 2014)
  - De-Mail Development (seit Jan 2014)
- Java, Groovy, Ruby bzw. Rails
- Middleware, Backend, Frontend

Über mich

Top 3 Fragen seit 1&1 Mitarbeiter

1. Du verkaufst DSL/Handy Verträge?
2. Kannst du mir ´nen billigen Vertrag besorgen?
3. Gibt es Marcell D‘Avis wirklich?

Code Review

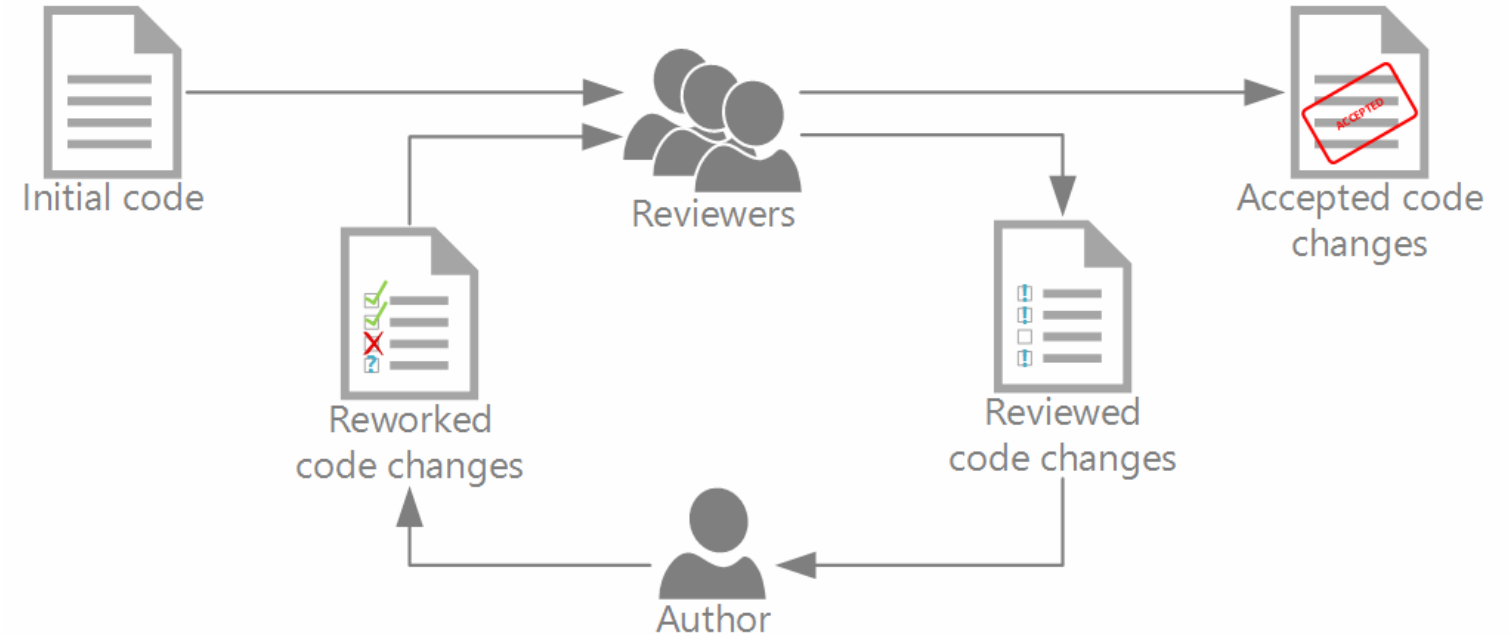
- „**Code review** is systematic examination of computer source code. It is intended to find mistakes overlooked in the initial development phase, improving the overall quality of software. Reviews are done in various forms such as pair programming, informal walkthroughs, and formal inspections.“ (Wikipedia)

- Norm für Code Reviews: **IEEE 1028**
  - Definition von verschiedenen Arten von Code Reviews und dem zugrundeliegenden Prozess
  - 1997 verabschiedet

Code Review - Varianten

- Pre-Commit Code Review
  - CR wird durchgeführt bevor Commit in Master
- Post-Commit Code Review
  - CR wird nach Submit in Master durchgeführt
- Pair-Programming
  - CR erfolgt parallel zur Entwicklung
- Mob Programming / Architecturing
  - Ganzes Entwicklerteam im Einsatz
  - 1 Tastatur
  - 1 Beamer

Code Review - Prozess



Code Beispiele (1)

```java
@Override
public Integer fetchIdentCount(Long provisioningId) throws UnknownProvisionError {
    LOG.info("MOCK: fetchIdentCount (provisioning_id = {}) called...", provisioningId);
    if (provisioningId < 0) {
        LOG.warn("MOCK: provisioning_id {} is unknown", provisioningId);
        throw new UnknownProvisionError("Unknown provisioning_id: " + provisioningId);
    }
    int count = 3;
    return count;
}
```

- unnötige Variablen Deklaration

Code Beispiele (2)

```
public SettingsPage() {
    // label to show current Date
    add(new Label("currentDate", new SimpleDateFormat("dd. MMMM yyyy", Locale.GERMAN).format(new Date())));

    UserInfo userInfo = authInfoService.getUserInfo();
    User user = acmServiceFacade.getUser();
    // label to show first and last name of user
    add(new Label("infoGreeting", userInfo.getFullname(true)));

    //Institution overview
    add(new InstitutionOverviewPanel("institutionOverviewPanel"));

    // label to show the religious name or pseudonym, in this case the kuenstlername
    String kuenstlername = userInfo.getReligiousnameOrPseudonym();
    Label kn = new Label("kuenstlername", kuenstlername);
    kn.setVisible(kuenstlername != null && !"".equals(kuenstlername));
    add(kn);
```

- Kommentare ohne Mehrwert

Code Beispiele (3)

```
private void init() {
    try {
        client = new DefaultRestClient("IdentRuby", "1");
        this.cancleUrl = ConfigurationUtil.readString(config,
                Id8OnlineClient.KEY_URL, true);
        String clientId = ConfigurationUtil.readString(config,Id8OnlineClient.KEY_CUSTOMER_ID, true);
        String clientPw = ConfigurationUtil.readString(config, Id8OnlineClient.KEY_CUSTOMER_CODE, true);

        URI build = UriBuilder.fromUri(cancleUrl.replace("{1}", StringUtils.EMPTY)).build();

        String basicAuthConfig = build.getHost() + ":" + httpPort + "," +clientId +":" +clientPw;
        client.setBasicAuthCredentialConfig(basicAuthConfig);

        this.useProxy = ConfigurationUtil.readBoolean(config,
                Id8OnlineClient.KEY_USE_PROXY, true);
        if (useProxy) {
            //..
            // set proxy
            /..
        }

        client.setBasicAuthCredentialConfig();
        client.setSoTimeout(socketTimeout);

    }catch(Exception e){
            close();
            throw new ConnectionException("Exception reading properties", e);
        }
}
```

public void setBasicAuthCredentialConfig(String... config)

Code Review – Warum?

- Abweichungen von definierten Standards, z.B. Verletzung von Namenskonventionen
- Fehler gegenüber den Anforderungen
- Fehler im Design
- schlechtes Design
- unzureichende Wartbarkeit
- Falsche Schnittstellenspezifikationen
- Schlecht geschriebener Code
- Fehlende Tests
- Falsche oder unnötige Kommentare
- Bugs frühzeitig erkennen

Code Review

* gilt nur für QA/Live

Code Review - Ziele

- “schöner“ Code
- fachlich korrekter Code
- Vermeidung von unnötigem Code (DRY, Clean Code)
- Vorbeugung von Bugs und Regressions
- Wartbarer Code

- Nebenläufig: Wissens-Transfer

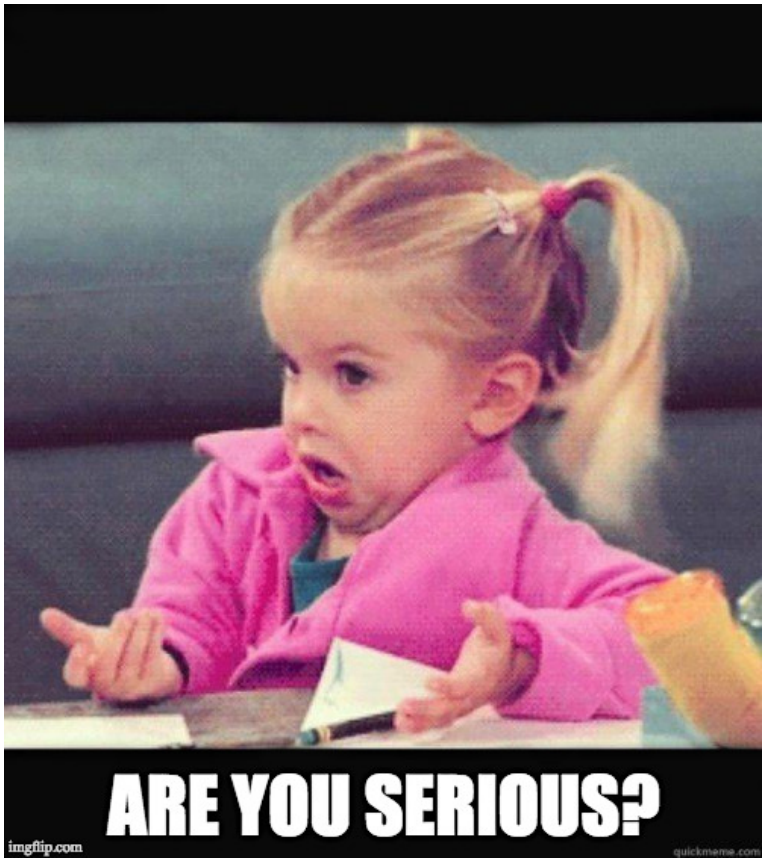
De-Mail Development

Code Review bei De-Mail Development

- 11 Entwickler
- 3 Systeme im Live/Production Betrieb:
  - De-Mail Leistungssystem
  - Beantragungsstrecke PK
  - Beantragunsstrecke KMU
- Programmiersprachen im Einsatz:
  - Java
  - Groovy
  - Ruby/Ruby on Rails
  - Java Script
- Frameworks/Fremdbibliotheken > 100
- Projekte in GIT: 43
- LOC > 450.000

Code Review bei De-Mail Development

Chef: „Im Team sollte jeder alles können...“



Code Review bei De-Mail Development

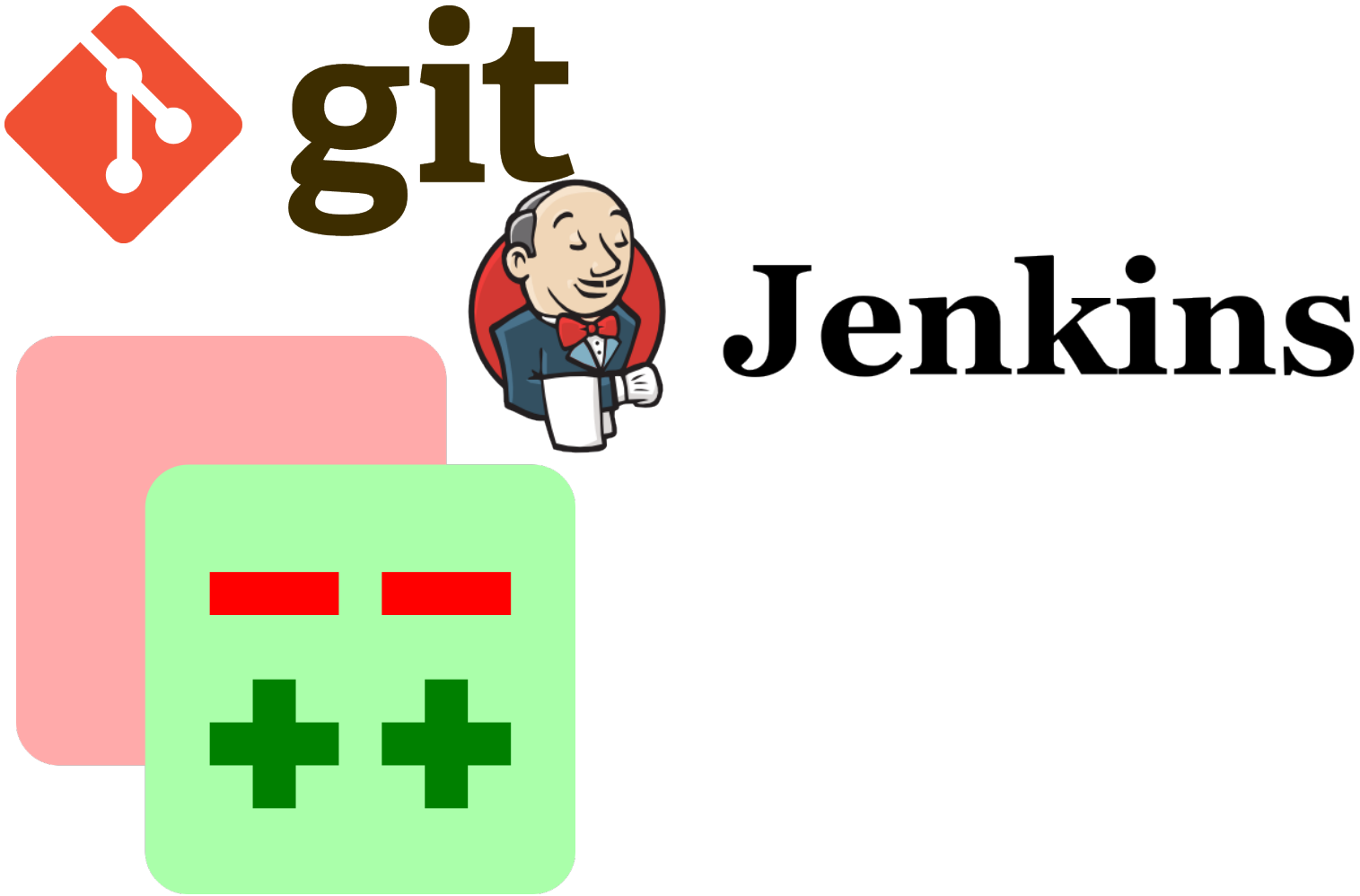
Code Review Pflicht!

=> keine Zeile ungesehen in Master

- Pair-Programming
- Pre-Commit Code Reviews

- Regeln:
  - CR muss von RDev durchgeführt werden
  - Pair Programming mit RDev erfordert keine CR

Tooling



Tooling – Gerrit im Detail

- Web basiertes Tool zur Code Review
- auf GIT Projekte anwendbar
- vorangeschaltetes Repository
- jeder Commit erzeugt Changeset
- Submit in den Master wird nach erfolgreichem Voting möglich
- Mögichkeit CI anzubinden (Jenkins/Hudson)

```
git push origin HEAD:refs/for/$branch_name
```

Tooling – Gerrit Voting System

- Code Review
  - -2 Do not submit
  - -1 I would prefer that you didn't submit this
  - 0 No score
  - +1 Looks good to me, but someone else must approve
  - +2 Looks good to me, approved
- Zusätzlich Jenkins:
  - -2 Build Failure
  - -1 Build Unstable
  - +1 Success
- Zum Submit wird +3 benötigt

Demo

Demo

So kann es aber auch kommen

Erfahrungen im Team

- weniger Bugs und Incidents
- Wissensaustausch enorm
- stabilere Builds
- sauberer/schönerer Code
- besseres Gefühl, gerade bei Bugs
- kontinuierliches Feedback durch Arbeitskollegen
- Freude über jedes „+2“ beim ersten Patchset

Code Reviews are awesome!!!

Probiert es doch einfach selbst aus!!

Das war‘s...

Danke

heiko.gramlich@1und1.de