Seminar Software Qualität

Auftaktveranstaltung

Maximilian Junker

Technische Universität München
Informatik IV: Software & Systems Engineering
Prof. Dr. Dr. h.c. Manfred Broy

Organisatorisches

Termin für das Seminar

Optionen: 22. Januar – 2. Februar

Bitte in Doodle eintragen:

https://doodle.com/poll/wn8ci39d88xq5uug

Termine

- Version 1.0 Seminararbeit: 2 Wochen vor Vortrag
- Probevortrag (verpflichtend): 2 Wochen vor Vortrag
- Finale Abgabe Seminararbeit: 1 Woche nach Vortrag

Agenda

1. Organisatorisches
2. **Software Qualität – Eine Motivation**
3. Effektiv Präsentieren
4. Literaturrecherche

Software Qualität

Eine Motivation

Maximilian Junker

(mit Folien von Benedikt Hauptmann, Henning Femmer, und Florian Deissenböck)

Technische Universität München

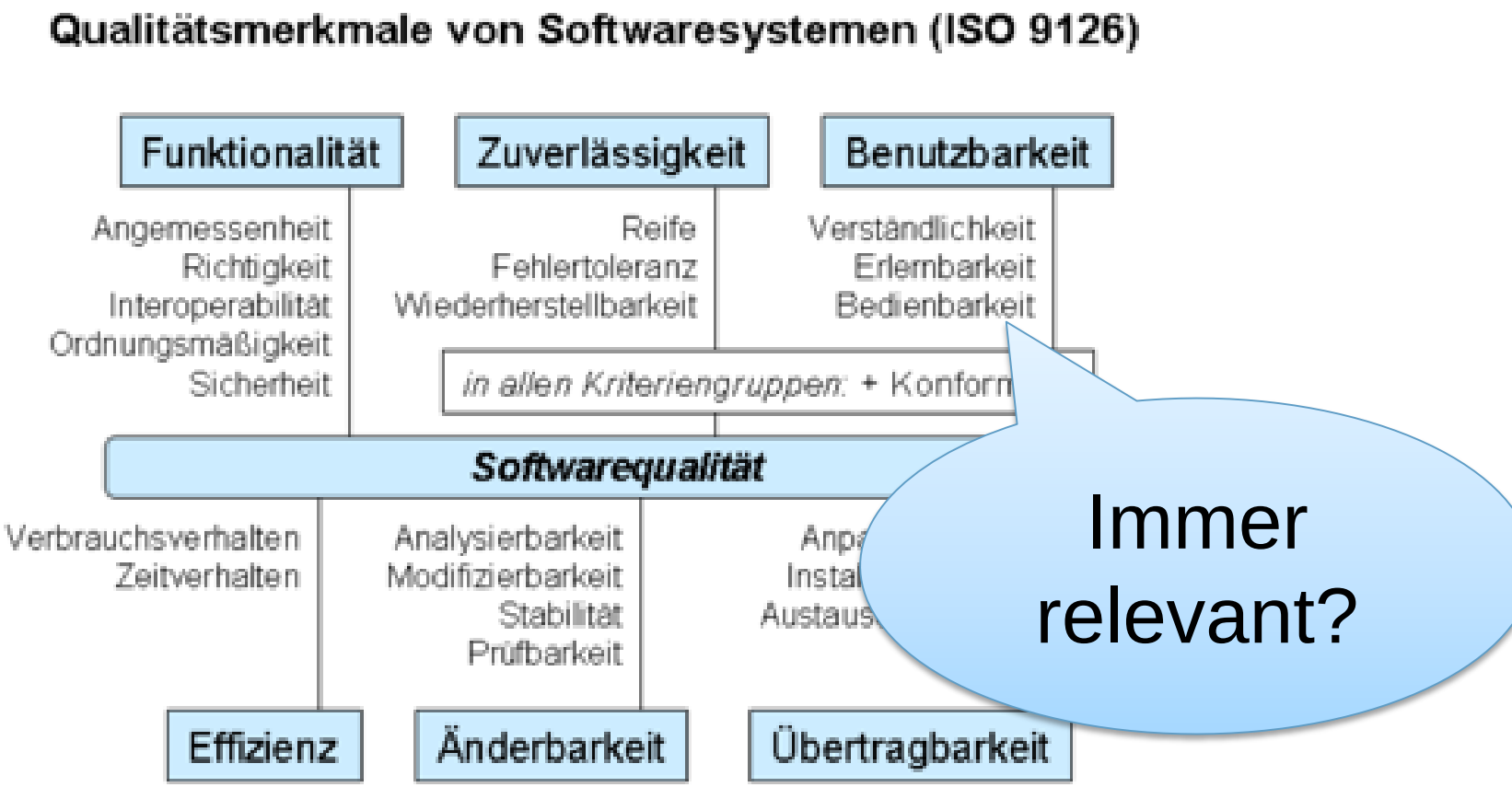
Informatik IV: Software & Systems Engineering

Prof. Dr. Dr. h.c. Manfred Broy

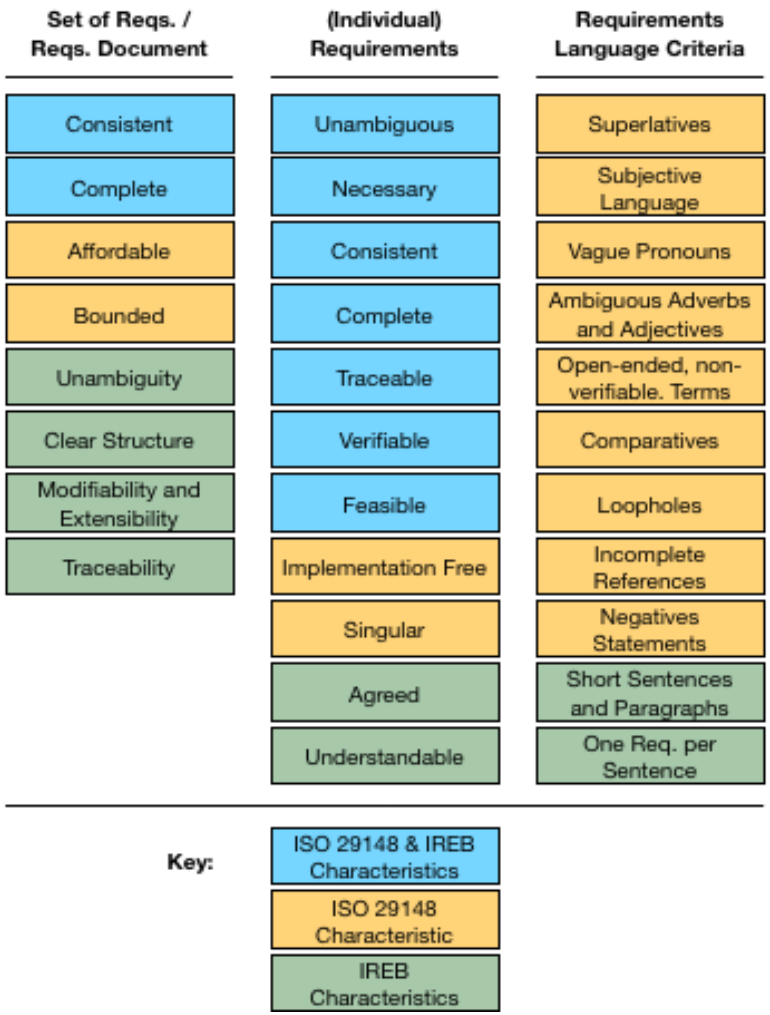
Agenda

1. Software Qualität?
2. Qualität von Sourcecode
3. Qualität von Anforderungen
4. Qualität von Tests
5. Tools für die Qualitätssicherung

Software Qualität fassen – Qualitätsmodelle

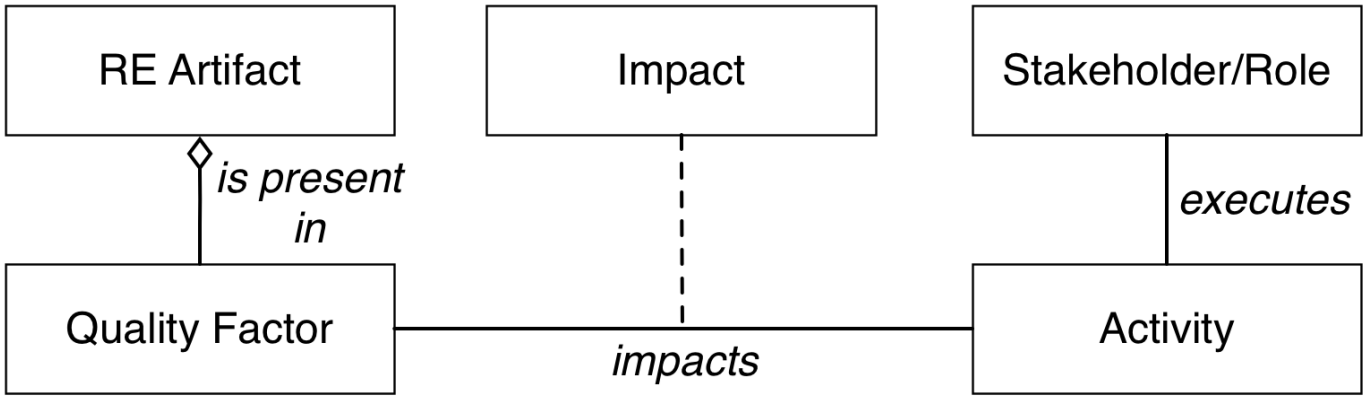


Unterschiedliche Qualitätsmodelle – Was nun?



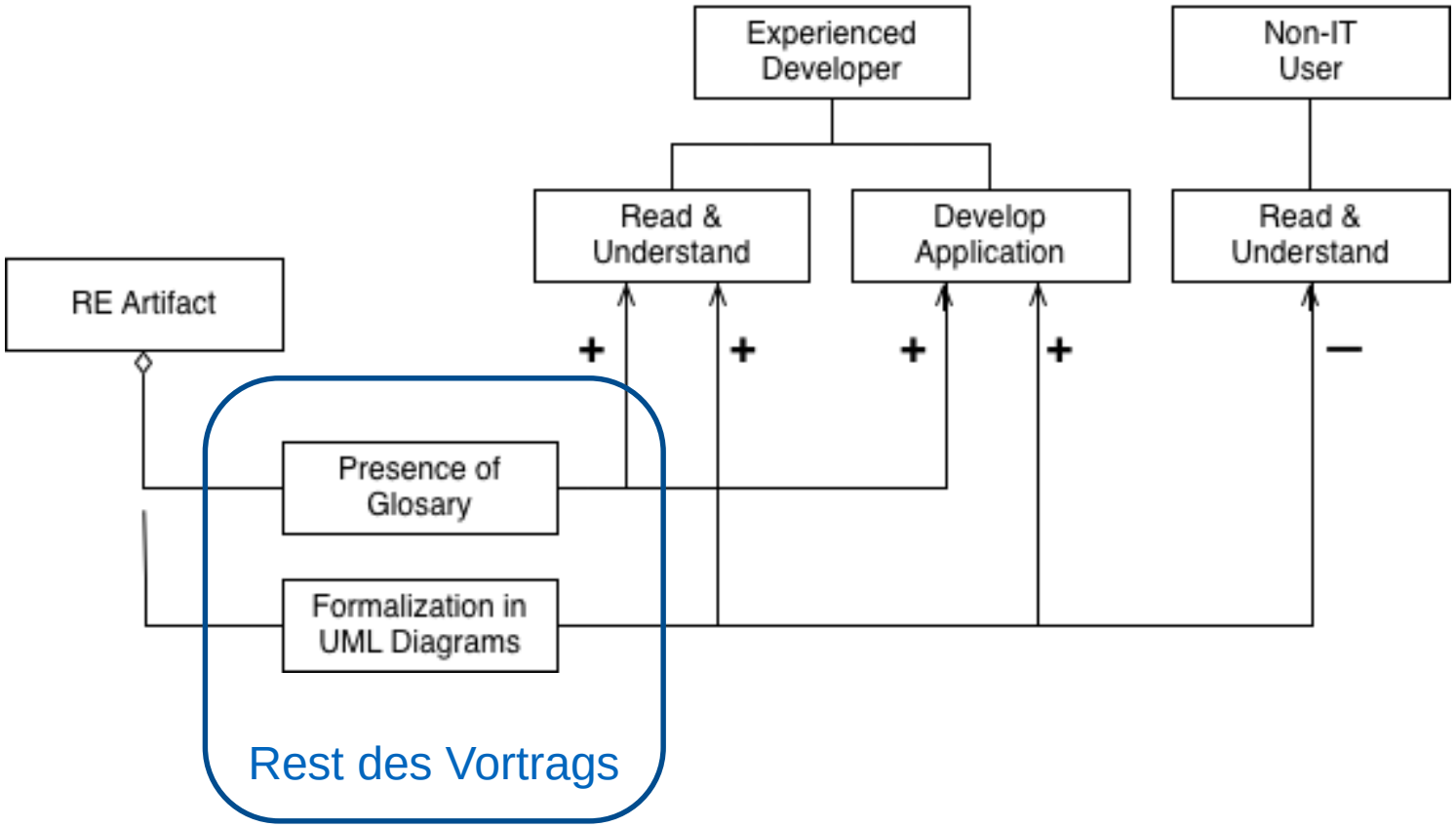
Aktivitätsorientierte Qualitätsmodelle

Beispiel Requirements Engineering

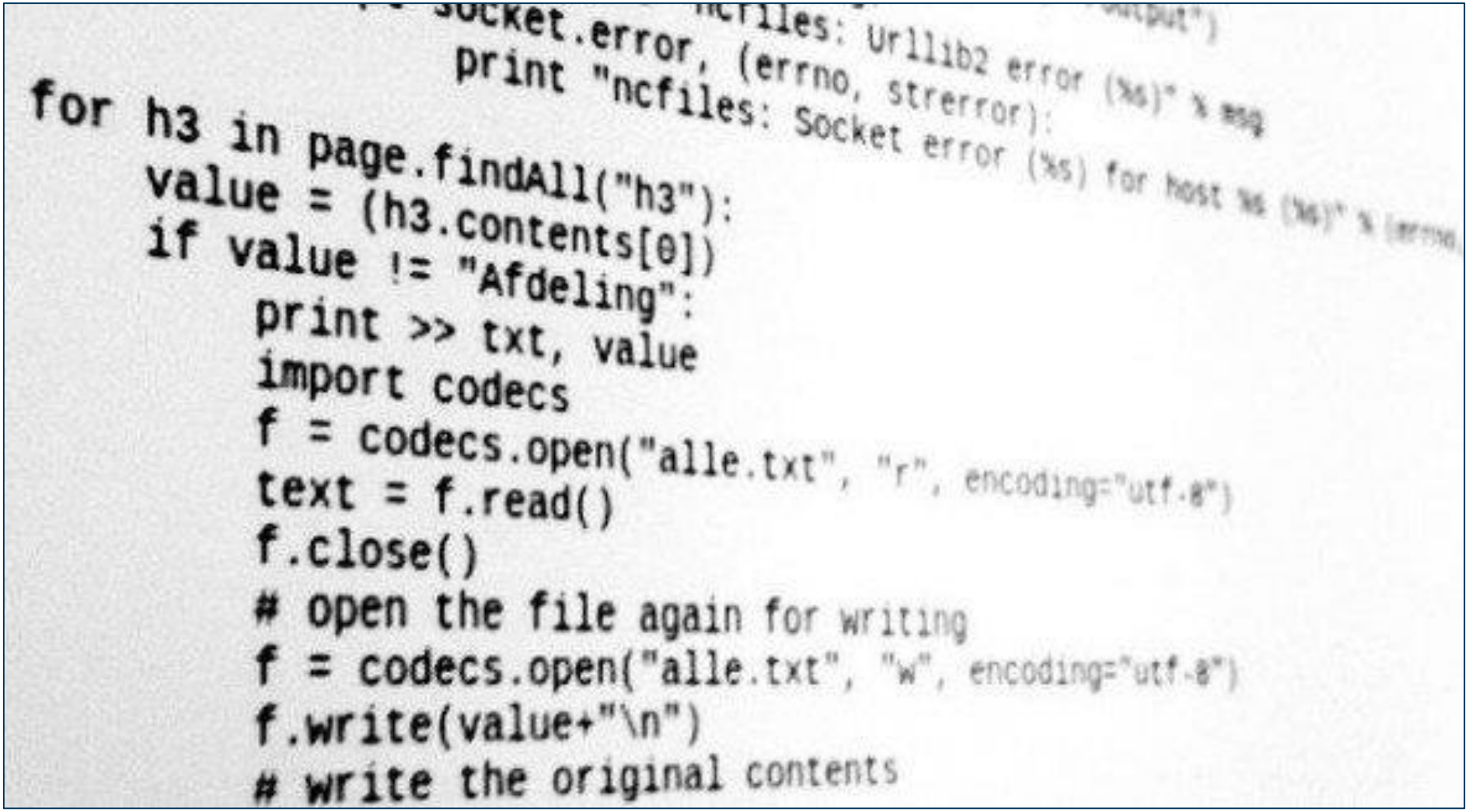


Aktivitätsorientierte Qualitätsmodelle

Beispiel Requirements Engineering

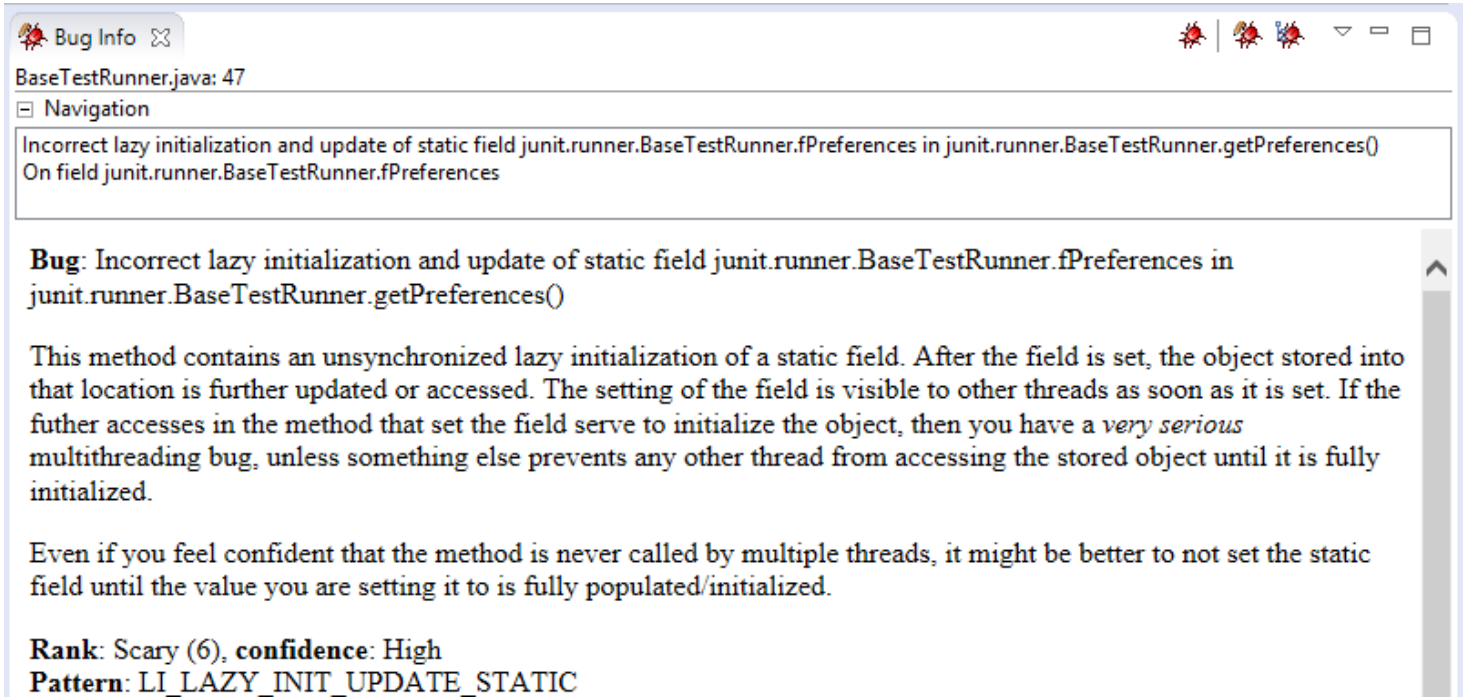


Qualität von Sourcecode



Potentielle Bugs

```
46      protected static Properties getPreferences() {
47          if (fPreferences == null) {
48              fPreferences = new Properties();
49              fPreferences.put("loading", "true");
50              fPreferences.put("filterstack", "true");
51              readPreferences();
52          }
53          return fPreferences;
54      }
```



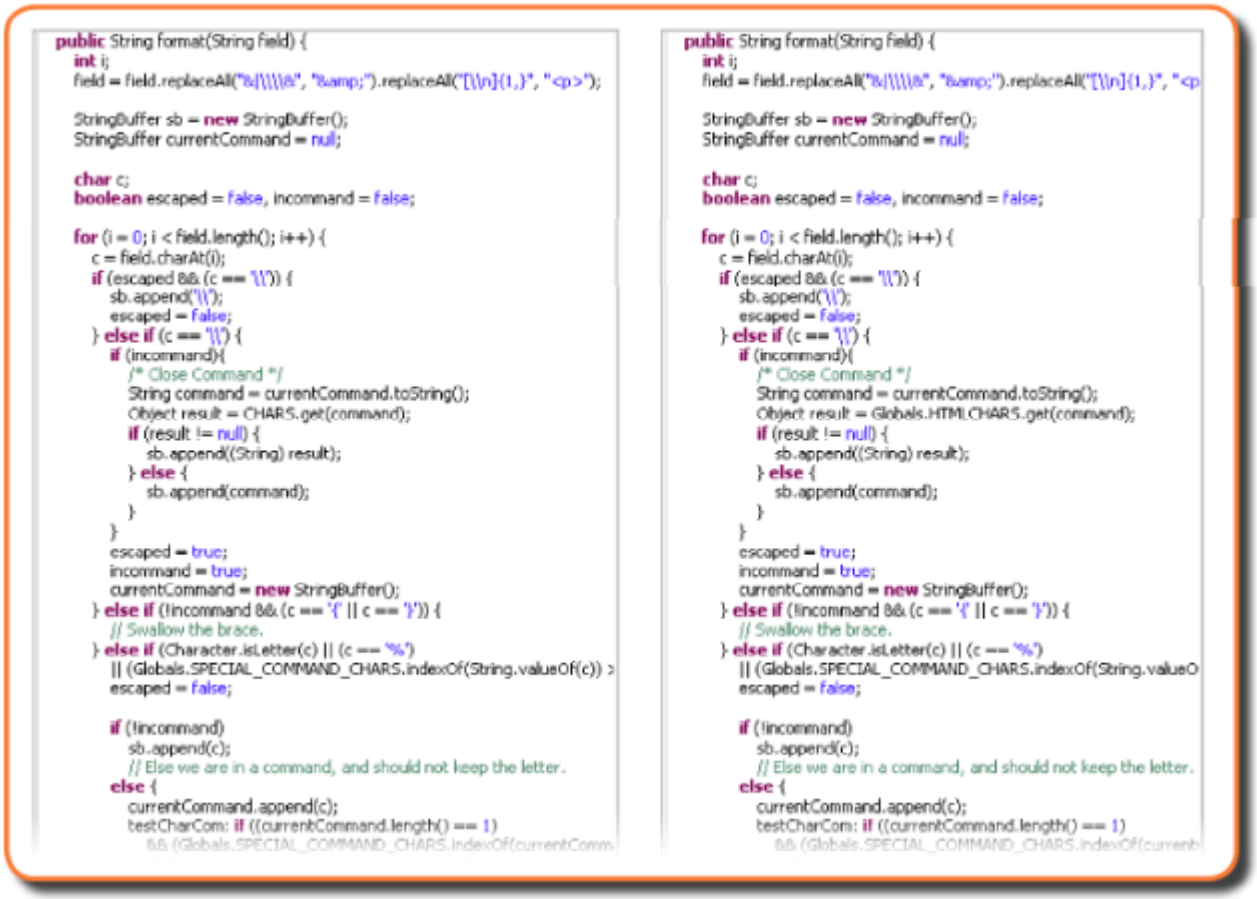
Formatierung

```
        hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
    hashOut.length = SSL_SHA1_DIGEST_LEN;
    if ((err = SSLFreeBuffer(&hashCtx)) != 0)
        goto fail;

    if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

        err = sslRawVerify(ctx,
                           ctx->peerPubKey,
                           dataToSign,                          /* plaintext */
                           dataToSignLen,                  /* plaintext length */
                           signature,
                           signatureLen);
        if(err) {
                sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                    "returned %d\n", (int)err);
                goto fail;
        }

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
```

Schachtelungstiefe

```
                for (const_iterator<Permission> permission =
                    permissions.begin();
                    permission != permissions.end(); ++permission) {
                   if (!permission->isInternal()) {
                      if (user->hasPermission (permission)) {
                         cout << "  " << permission << endl;
                      }
                   }
                }
             }
          }
       } /* bezieht sich auf if in Zeile 172 */
}
```

Klone



```
public String format(String field) {
   int i;
   field = field.replaceAll("&|\\\\|&", "&amp;").replaceAll("[\\n]{1,}", "<p>");

   StringBuffer sb = new StringBuffer();
   StringBuffer currentCommand = null;

   char c;
   boolean escaped = false, incommand = false;

   for (i = 0; i < field.length(); i++) {
      c = field.charAt(i);
      if (escaped && (c == '\\')) {
         sb.append('\\');
         escaped = false;
      } else if (c == '\\') {
         if (incommand){
            /* Close Command */
            String command = currentCommand.toString();
            Object result = CHARS.get(command);
            if (result != null) {
               sb.append((String) result);
            } else {
               sb.append(command);
            }
         }
         escaped = true;
         incommand = true;
         currentCommand = new StringBuffer();
      } else if (!incommand && (c == '{' || c == '}')) {
         // Swallow the brace.
      } else if (Character.isLetter(c) || (c == '%')
         || (Globals.SPECIAL_COMMAND_CHARS.indexOf(String.valueOf(c)) >
         escaped = false;

         if (!incommand)
            sb.append(c);
            // Else we are in a command, and should not keep the letter.
         else {
            currentCommand.append(c);
            testCharCom: if ((currentCommand.length() == 1)
               && (Globals.SPECIAL_COMMAND_CHARS.indexOf(currentComm
```

```
public String format(String field) {
   int i;
   field = field.replaceAll("&|\\\\|&", "&amp;").replaceAll("[\\n]{1,}", "<p

   StringBuffer sb = new StringBuffer();
   StringBuffer currentCommand = null;

   char c;
   boolean escaped = false, incommand = false;

   for (i = 0; i < field.length(); i++) {
      c = field.charAt(i);
      if (escaped && (c == '\\')) {
         sb.append('\\');
         escaped = false;
      } else if (c == '\\') {
         if (incommand){
            /* Close Command */
            String command = currentCommand.toString();
            Object result = Globals.HTMLCHARS.get(command);
            if (result != null) {
               sb.append((String) result);
            } else {
               sb.append(command);
            }
         }
         escaped = true;
         incommand = true;
         currentCommand = new StringBuffer();
      } else if (!incommand && (c == '{' || c == '}')) {
         // Swallow the brace.
      } else if (Character.isLetter(c) || (c == '%')
         || (Globals.SPECIAL_COMMAND_CHARS.indexOf(String.valueO
         escaped = false;

         if (!incommand)
            sb.append(c);
            // Else we are in a command, and should not keep the letter.
         else {
            currentCommand.append(c);
            testCharCom: if ((currentCommand.length() == 1)
               && (Globals.SPECIAL_COMMAND_CHARS.indexOf(currenth
```

Anforderungsqualität

Vage Phrasen

The system generates relations to other files, based on attribute values.

The controls shall be illuminated where appropriate.

The device shall allow the user to request recommendations in as few steps as possible.

Undefinierte Akronyme

ERTMS/ETCS	The ETCS part of ERTMS
MA	Movement Authority
RBC	Radio Block Centre

The ERTMS/ETCS on-board equipment shall send an MA request to the RBC and wait. If an SR authorisation is received from RBC, the process shall go to S24.

?

Unvollständige Anforderungen

Platzhalter und Standardtexte

[This section contains the preconditions of the use case.]

tbd.

Klone

Copy & Paste Reuse (*Cloning*) führt zu

1. Höheren Wartungsaufwänden
2. Inkonsistenzen

[REQ101]
Component A must receive data and confirmation.

[REQ201]
Component B must receive data and confirmation.

•••

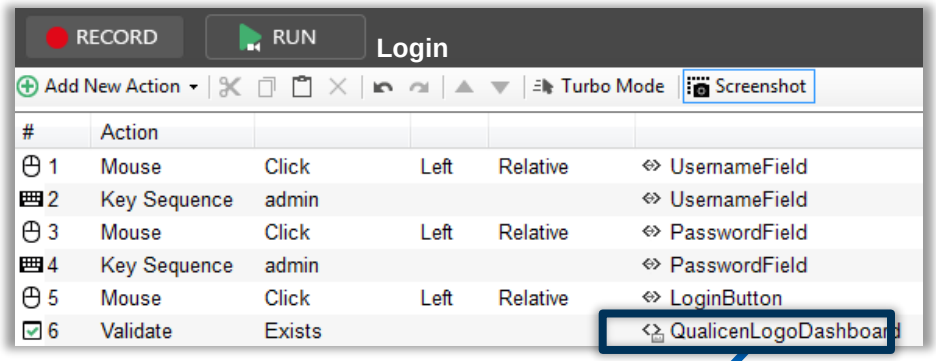
[REQ801]
Component C must receive data and confirmation.

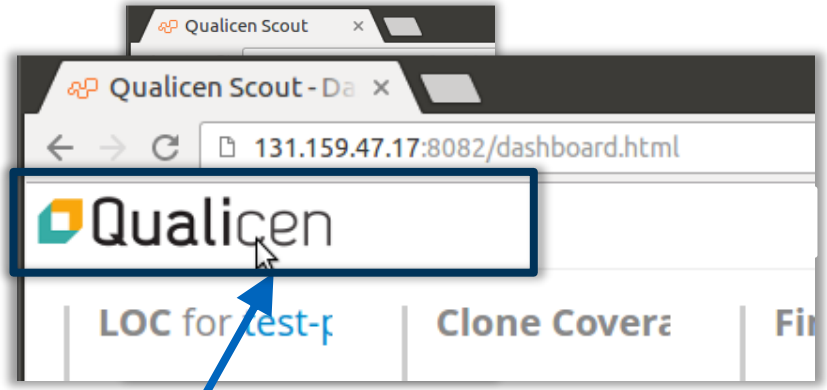
[REQ401]
The component D must receive data and confirmation.

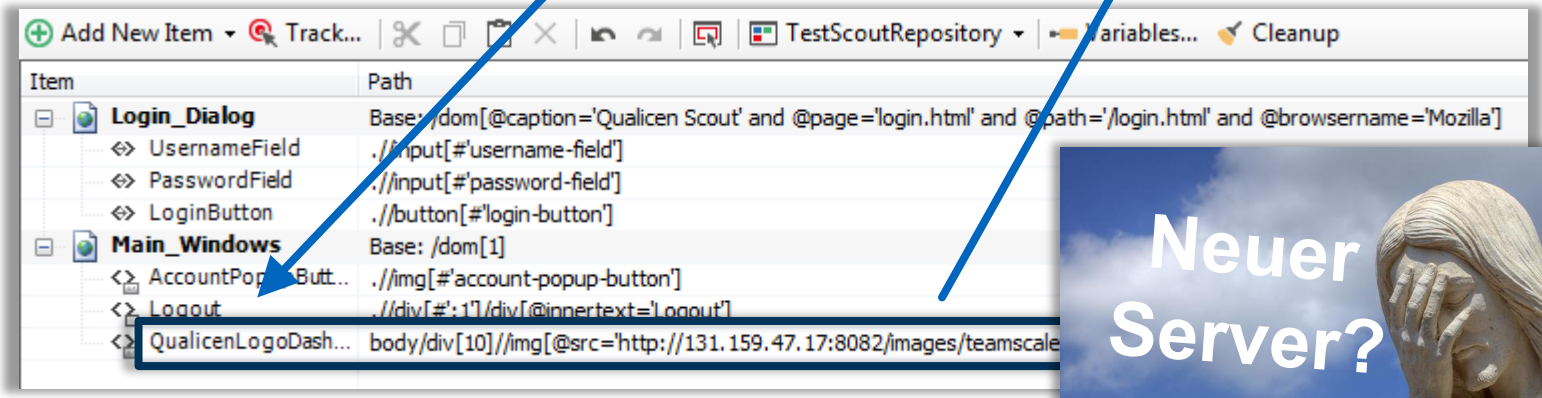
[REQ501]
Component E must receive **sender name** , data, and confirmation.

Testqualität

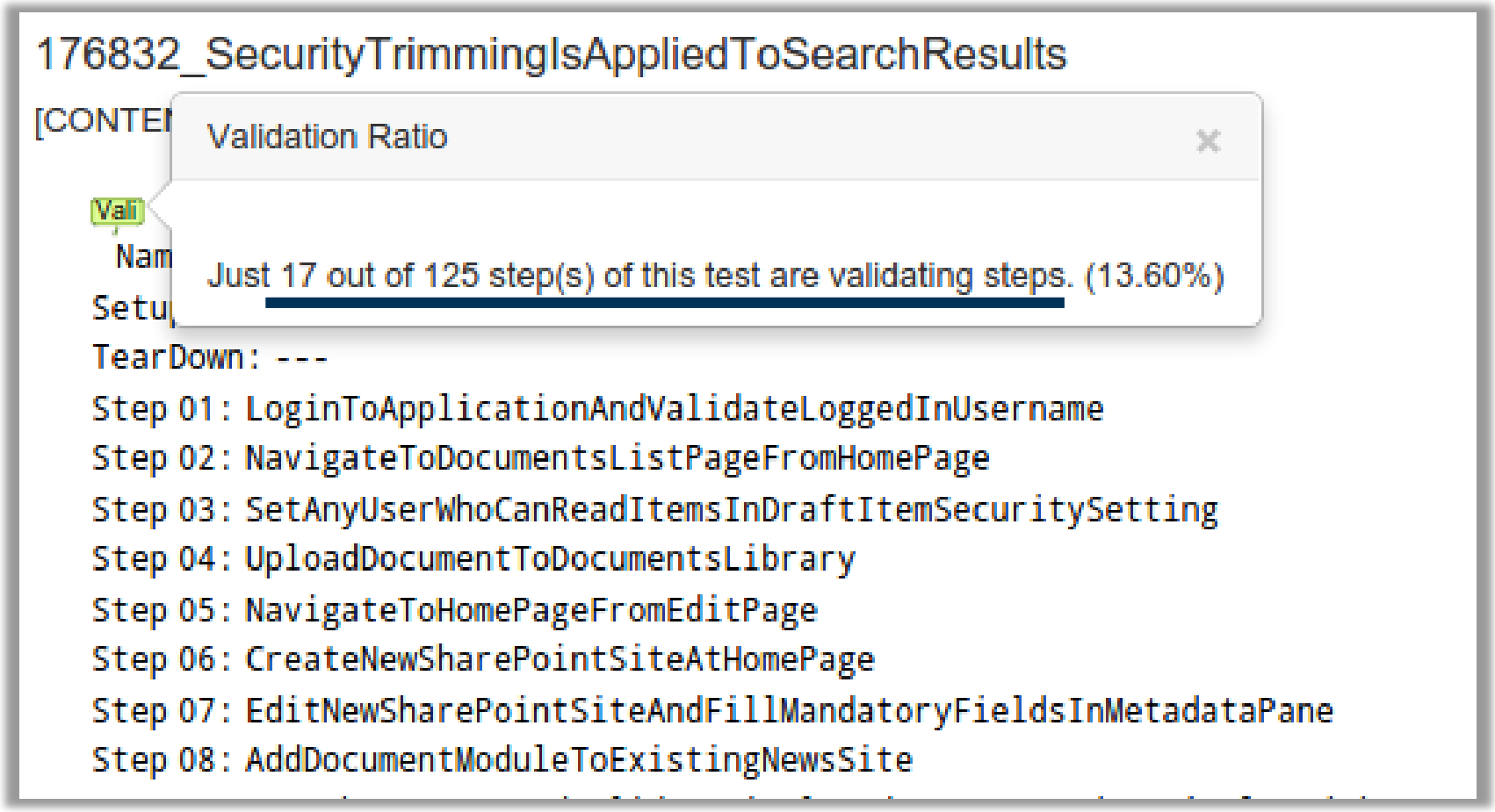
Fragile Tests



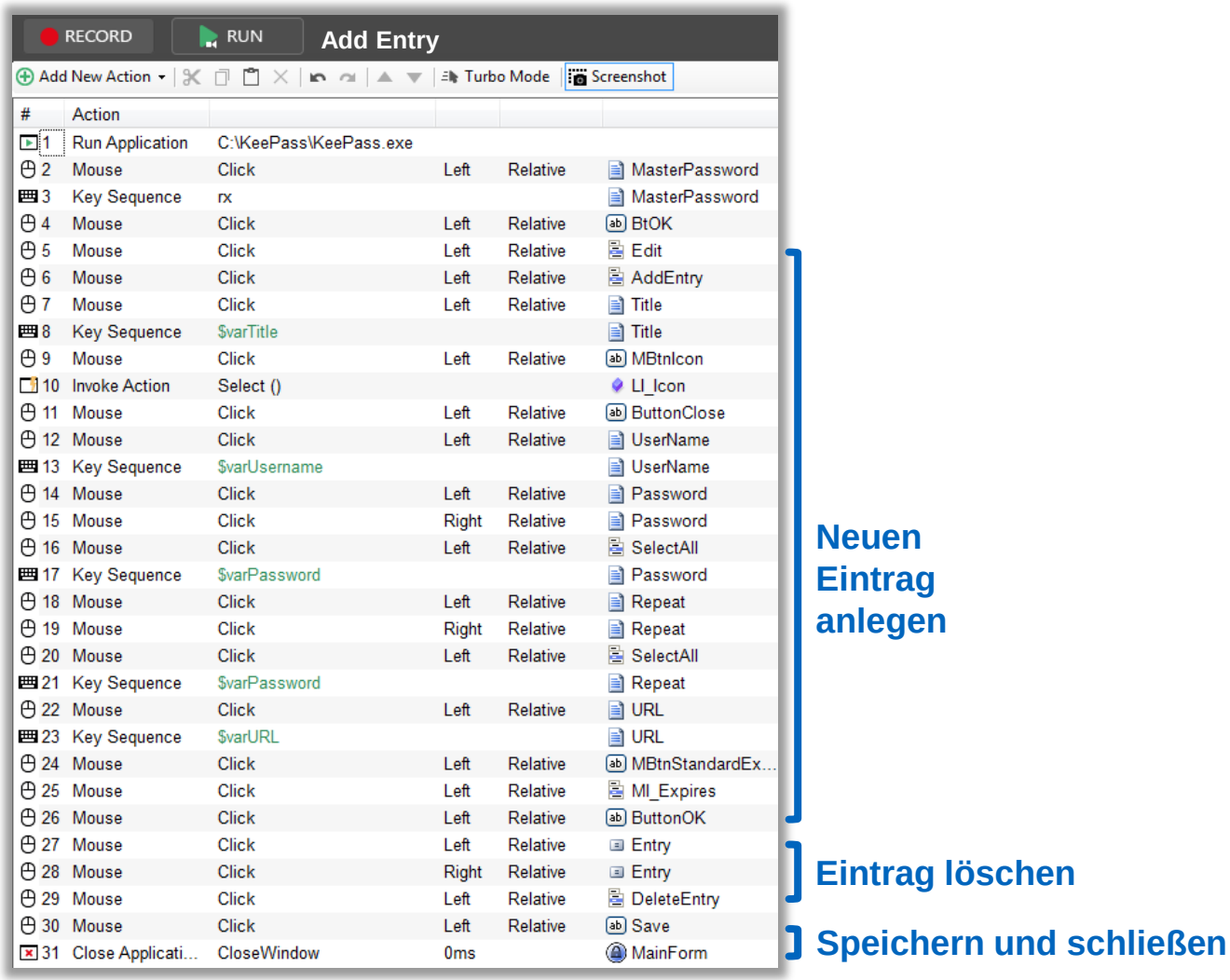




Ineffektive Tests



Keine oder schlechte Struktur in Tests



Tools für die Qualitätssicherung

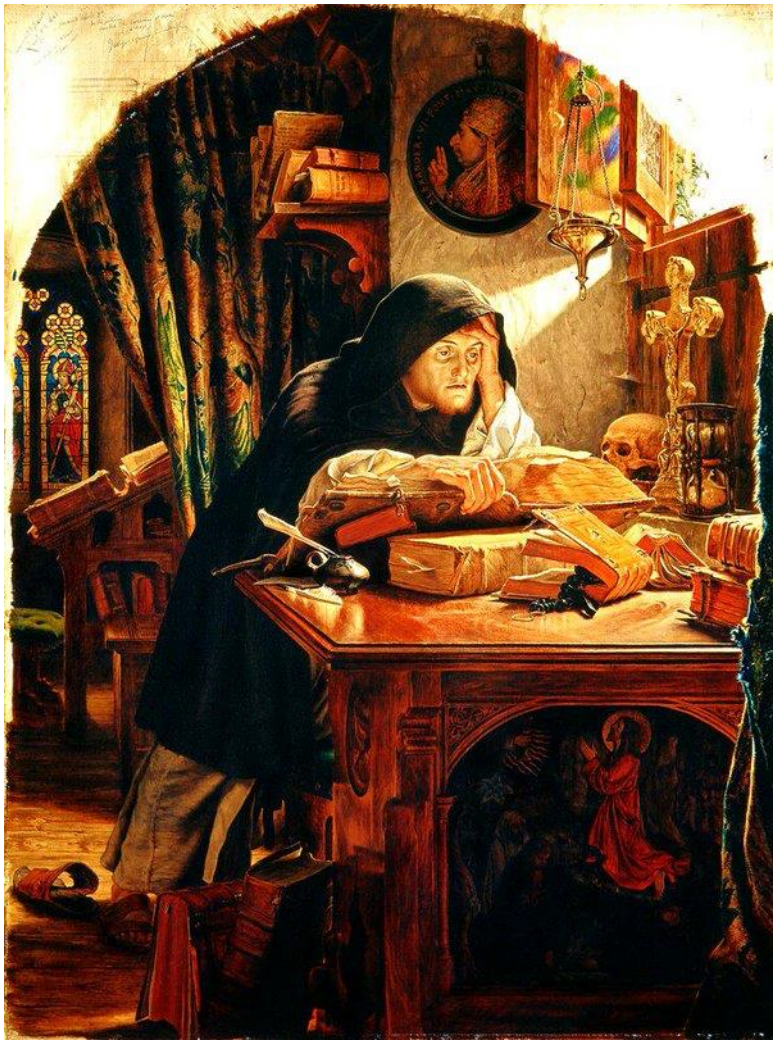
Manuelle Reviews

Effektiv:

- Genaue inhaltliche Prüfung möglich
- Prüfen schwer automatisch messbare subjektive Verständlichkeit
- Wissensvermittlung

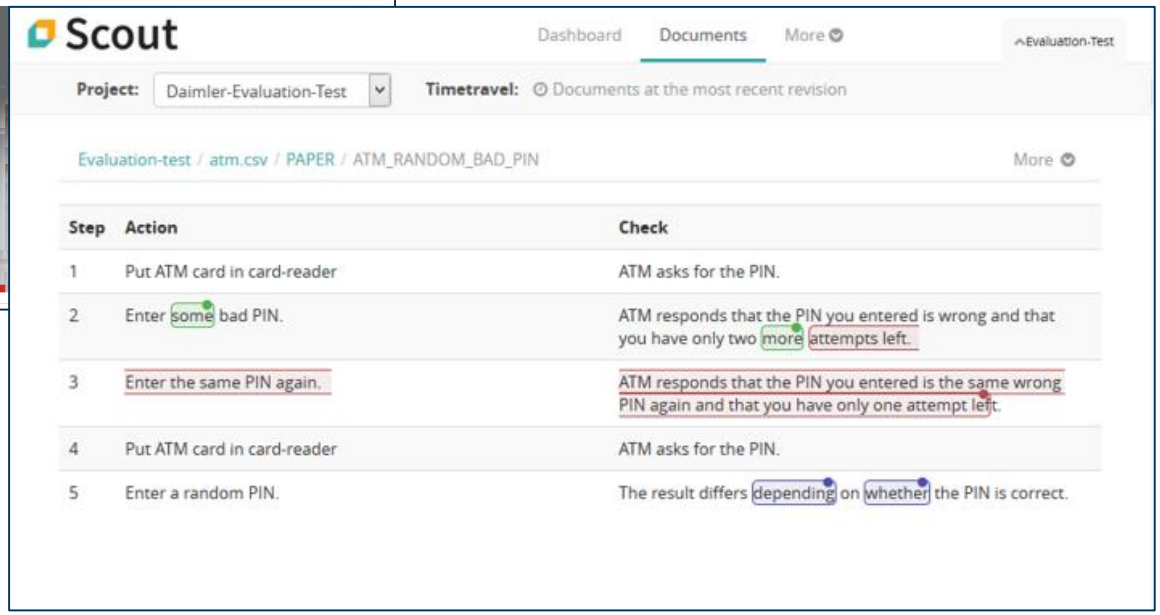
Aber:

- Aufwendig und teuer.
- Lange Feedback-Schleifen.
- Inkonsistent.

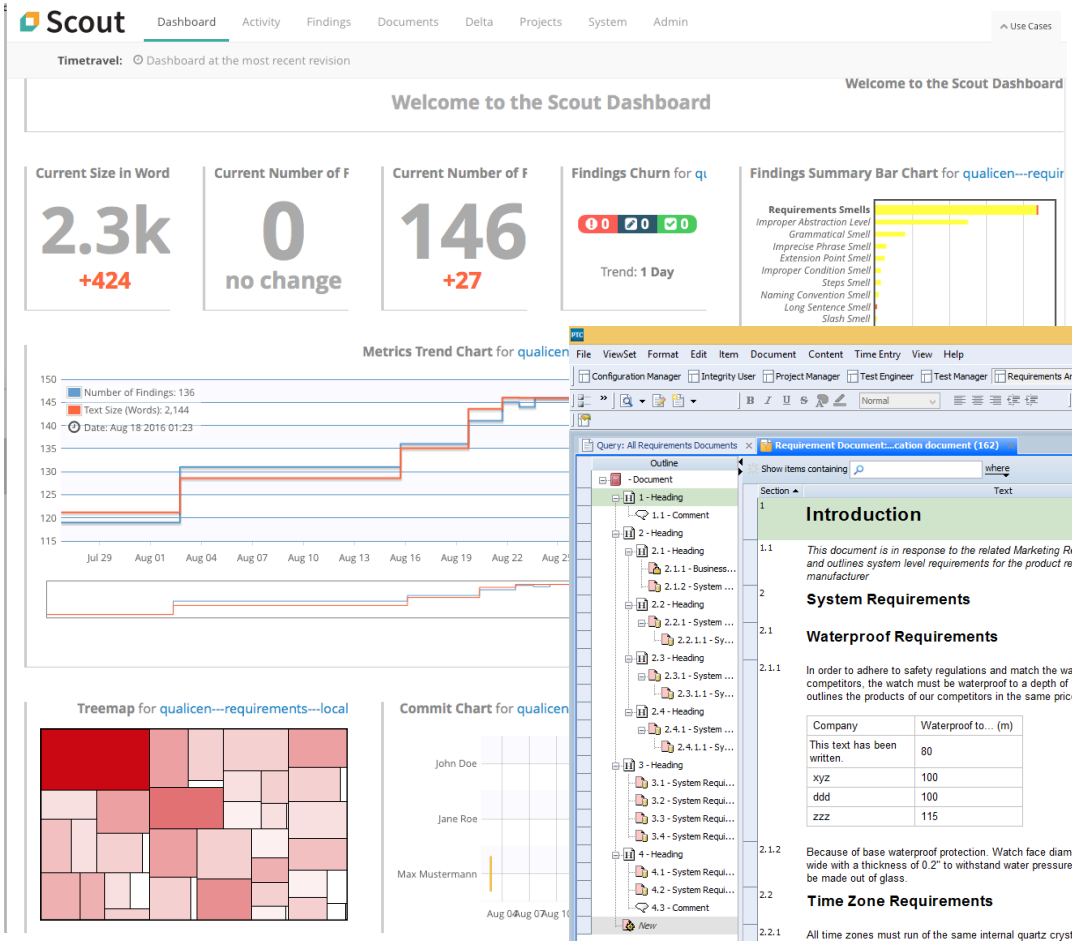


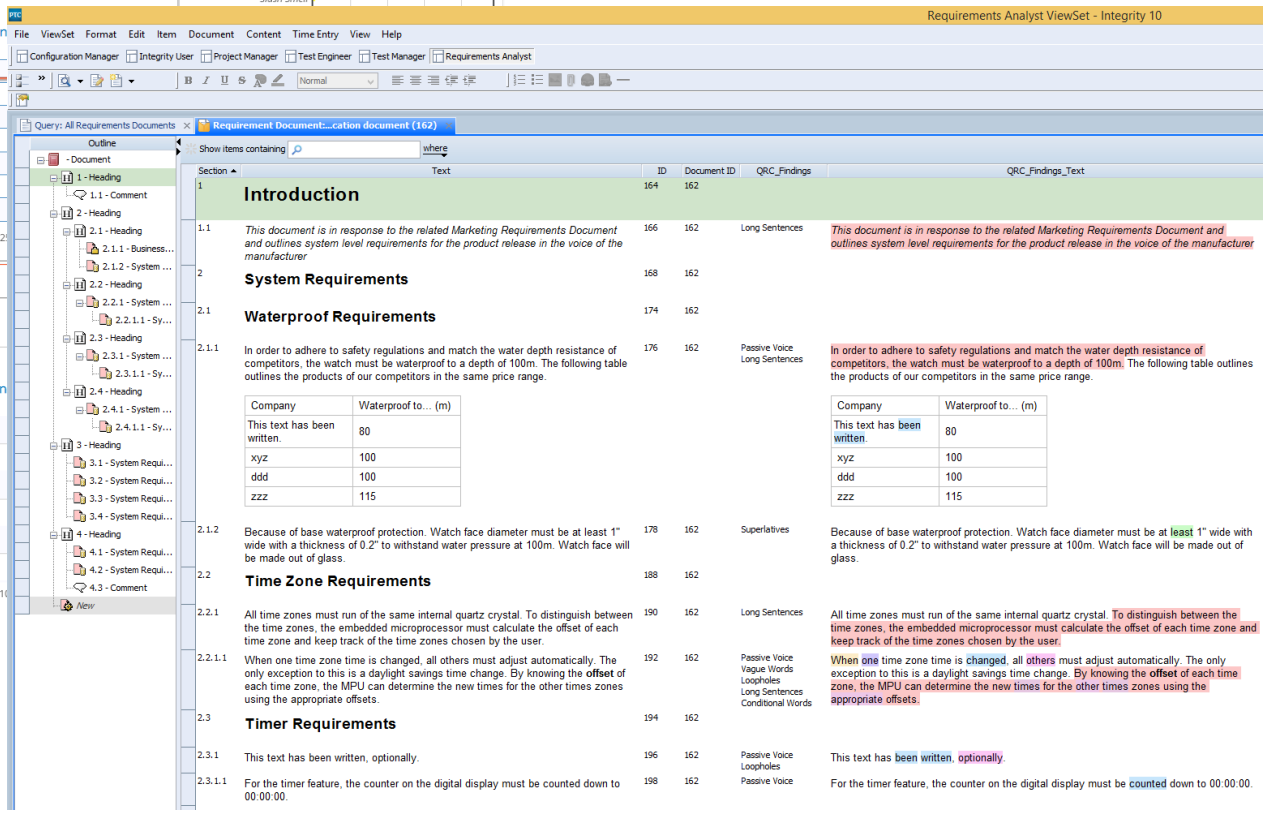
Automatisierte Qualitätsanalyse

Teamscale und Scout



Automatische Qualitätsanalyse – Zwei Perspektiven





Automatische oder manuelle Qualitätssicherung?

- **Viele relevante Qualitätseigenschaften sind nicht automatisch analysierbar**
  - Kommentare
  - sinnvolle Verwendung von Datenstrukturen und Algorithmen
  - Inhalte von Anforderungen
  - logische Redundanz
- **Diese Eigenschaften bedürfen der manuellen Analyse**
- **Manuelle Analysen sollten soweit möglich durch automatische Analysen unterstützt werden**