Survey - Practitioner's Perspective on the Readability of Log Messages

Dear Participant,

We are conducting a survey on what are practitioners' expectations for the readability of log messages and how to improve it. This survey will take you around 15 minutes.

Software logs are widely used in practice to assist in many tasks (e.g., debugging, testing). Such logs are generated by logging statements written by the developers. Below is an example logging statement from Apache Flink. In our study, we investigate the readability of log messages in the logging statement (i.e., the sentence in **green** in the example below).

```
…
} catch (Exception e) {
        LOG.warn( "Cannot delete closed and discarded state stream for {}.", statePath, e);
…
```

There exist some partial guidelines on composing logging statements (e.g., how to choose the log level). However, to the best of our knowledge, there is still a lack of well-defined guidelines on writing the **log message**. Therefore, how to write log messages with good readability that can clearly and sufficiently record system runtime behaviors is still challenging.

As such, we have conducted a series of interviews with industrial practitioners to investigate their expectations for the readability of log messages and what practices can be taken to improve it. Based on the interview results, we derived three aspects related to the readability of log messages and several practices to improve on each aspect.

The main goal of this survey is to: (1) investigate your opinion on the findings derived from our interviews with industrial practitioners; (2) examine the readability of several real-world log messages; (3) collect your other comments and feedback on our study.

We will not reveal your identity in any form of the research outcomes. All of the information provided in the survey will be used for research purposes only. We sincerely appreciate your participation in our study.

Part 1: Information about the Participant

In this part, we will ask you for some background information related to your role and experience.
For each question, please type your answer in the corresponding area.

Q1: Which of the following best describes your primary job role?
A. Software engineer B. Test engineer C. Project manager
D. Researcher/Research engineer E. Data analyst F. Other (Please specify below)
Answer:

Q2: How many years of experience do you have in software development/software maintenance/programming?
Answer:

Part 2: Aspects of the Readability of Log Messages

From our prior interviews with practitioners, we derive three aspects that are related to the readability of log messages: Structure, Information, and Wording. We will illustrate each aspect by providing you with two real-world examples which are against and for the readability in the corresponding aspect, respectively.
For each question, please type your answer in the corresponding area.

Aspect 1 - Structure: the format and organization of words and variables that a log message presents its information.

Example log message against the Structure aspect of readability:

```
…
logger.debug("Bootstrap variables: {} {} {} {}",
        DatabaseDescriptor.isAutoBootstrap(),
        SystemKeyspace.bootstrapInProgress(),
        SystemKeyspace.bootstrapComplete(),
        DatabaseDescriptor.getSeeds().contains(FBUtilities.getBroadcastAddress()));
…
```

Note: four variables are directly presented one by one, might be difficult to distinguish the meaning of each variable.

Example log message for the Structure aspect of readability:

```
…
logger.debug("[repair #{}] Repair completed between {} and {} on {}", getId(), nodes.endpoint1,
                                    nodes.endpoint2, desc.columnFamily);
…
```

Q3: How do you consider the importance of Structure to the readability of log messages?
A. Very important B. Important C. Neutral D. Unimportant E. Very unimportant F. Not sure
Answer:
Comment: Optional, please remove this sentence and leave your comments here, same for the following comments

Aspect 2 - Information: the semantic information conveyed by the log message to record system execution behaviors.

Example log message against the Information aspect of readability:

```
…
LOG.info("Started.");
…
```

Example log message for the Information aspect of readability:

```
…
LOG.info("Quota support disabled, not starting space quota manager.");
…
```

Q4: How do you consider the importance of Information to the readability of log messages?

A. Very important B. Important C. Neutral D. Unimportant E. Very unimportant F. Not sure

Answer:

Comment:

Aspect 3 - Wording: the lexical usage of words and punctuation in the log message.

Example log message against the Wording aspect of readability:

```
…
LOG.info("Added to offline, CURRENTLY NEVER CLEARED!!! ");
…
```

Note: This log message uses very emotional wording to present a normal event.

Example log message for the Wording aspect of readability:

```
…
LOG.info("No family specified, will execute for all families.");
…
```

Note: Different from the prior one, this log message uses standard wording.

Q5: How do you consider the importance of Wording to the readability of log messages?

A. Very important B. Important C. Neutral D. Unimportant E. Very unimportant F. Not sure

Answer:

Comment:

Q6: Overall, what is your perspective on these three aspects of the readability of log messages (Do these three aspects can reflect the readability of log messages)? Please only consider the readability of the **log messages**, other parts in the logs (e.g., timestamp, log level, the configuration of logging libraries) are not the focus of this study.

A. Very positive B. Positive C. Neutral D. Negative E. Very negative F. Not sure

Answer:

Comment:

Part 3: Practices on Improving the Readability of Log Messages

*For each aspect of the readability of log messages, we derived several specific practices for improving the readability of log messages from our prior interviews with practitioners. Some of the practices are "**corrective practices**", which are practices to improve the insufficiency of readability in log messages. Some of the practices are "**enhancing practices**", where developers can decide whether to apply them or not based on the situations and needs. We will illustrate each practice using an example with and/or without the corresponding practice. For each question, please specify your agreement level for the statement in the question title.*

Structure Practice 1 (Corrective): Have clear boundaries and distinctions among items.

Example 1 (without this practice):

```
…
LOG.debug("Reading from {} {} {} {}", tableDesc.getTableName(), region.getRegionNameAsString(),
                column.getNameAsString(), Bytes.toStringBinary(startKey));
…
```

Example 2 (with this practice):

```
…
LOG.debug("Reading from table: {}, region: {}, column: {}, key: {}", tableDesc.getTableName(),
region.getRegionNameAsString(), column.getNameAsString(), Bytes.toStringBinary(startKey));
…
```

Q7: This practice can improve the readability of log messages from the aspect of Structure.
A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Structure Practice 2 (Enhancing): Use an easy-to-parse structure if needed and possible

Example (with this practice):

```
…
logger.info("Summary of the change, term: {}, version: {}, reason: {}", newClusterState.term(),
                                    newClusterState.version(), task.source);
…
```

Note: Consider formatting the log message that is easy to be parsed by scripts for further analysis. Such as the example here uses a comma (“,”) to separate each part. The ideal purpose is to have log messages that are both human-readable and machine-readable.

Q8: This practice can improve the readability of log messages from the aspect of Structure.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Structure Practice 3 (Enhancing): Use parameterized logging to present the variables.

Example 1 (without this practice):

```
…
LOG.error("Exception when formatting: '" + dateStr + "' from: '" + fromFormat + "' to: '" + toFormat
                                                    + "'", e);
…
```

Example 2 (with this practice):

```
…
logger.info("Exception when formatting: '{}' from '{}' to '{}' ", dateStr, fromFormat, toFormat, e);
…
```

Note: The log message in the logging statement with parameterized logging is easier to revisit and revise. Moreover, though not related to readability, parameterized logging has better performance than simply concatenating the strings (according to the documentation of Log4J).

Q9: This practice can improve the readability of log messages from the aspect of Structure.
A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Information Practice 1 (Corrective): Provide proper context for the run-time behaviors.

Example 1 (without this practice):

```
…
Thread.currentThread().interrupt();
LOG.info("Interrupted");
…
```

Example 2 (with this practice):

```
…
Thread.currentThread().interrupt();
LOG.info("The current thread is interrupted"); //(also add thread ID if available)
…
```

Q10: This practice can improve the readability of log messages from the aspect of Information.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Information Practice 2 (Corrective): Write a self-explanatory log message that is independent of other log messages.

Example 1 (without this practice):

```
…
} catch (final AmazonClientException e) {
logger.info("Exception while retrieving instance list from AWS API: {}", e.getMessage());
logger.debug("Full exception:", e); //depending on the prior info log
…
```

Note: These two info and debug logs may not always be generated closely together. If there are other logs that appear before the debug log, it will be confusing to only see a "Full exception".

Example 2 (with this practice):

```
…
} catch (final AmazonClientException e) {
logger.info("Exception while retrieving instance list from AWS API: {}", e.getMessage());
logger.debug("Exception while retrieving instance list from AWS API, full exception: ", e);
…
```

Q11: This practice can improve the readability of log messages from the aspect of Information.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

<u>Information Practice 3 (Enhancing)</u>: Minimize noise, emphasize the key information.

Example 1 (without this practice):

```
…
LOG.warn("An HTTP error response in WebSocket communication would not be processed by the browser! If you need to send the error code and message to the client then configure custom WebSocketResponse via WebSocketSettings#newWebSocketResponse() factory method and override #sendError() method to write them in an appropriate format for your application. The ignored error code is '{}' and the message: '{}'.", sc, msg);
…
```

Example 2 (with this practice):

```
…
LOG.warn("An HTTP error response in WebSocket communication would not be processed by the browser, the ignored error code: '{}', the message: '{}'. ", sc, msg);
//mention the key information first, can add another log, or use another way to write the additional instruction if it's necessary
…
```

Q12: This practice can improve the readability of log messages from the aspect of Information.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Wording Practice 1 (Corrective): Use standard English words (e.g., avoid typos, incomplete words).

Example (without this practice):

```
…
LOG.debug("Failed to preform reroute after cluster settings were updated.");
…
//"preform" is a typo and should be "perform"
```

Q13: This practice can improve the readability of log messages from the aspect of Wording.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Wording Practice 2 (Corrective): Follow the convention of written language (e.g., correct grammar, not too oral).

Example (without this practice):

```
…
LOG.debug("Pinging a master {} but we do not exists on it, act as if its master failure");
…
//we do not “exists” should be “exist”
```

Q14: This practice can improve the readability of log messages from the aspect of Wording.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Wording Practice 3 (Corrective): Try to use impartial and neutral wording (e.g., avoid being too emotional or abusing capitalization).

Example 1 (without this practice):

```
…
LOG.error("!!!!!Uh-oh, didn't find any action handlers!!!!!");
…
```

Example 2 (without this practice):

```
…
LOG.info("Added to offline, CURRENTLY NEVER CLEARED!!! ");
…
```

Note: These two examples are too emotional. Should change to neutral wording.

Q15: This practice can improve the readability of log messages from the aspect of Wording.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Wording Practice 4 (Enhancing): Be careful on using Abbreviations & Acronyms.

Example:

```
…
LOG.warn("No TGT found: will try again at {}");
…
```

Note: Developers should ensure that the user can understand the meaning of abbreviations and acronyms before writing them into the log message.

Q16: This practice can improve the readability of log messages from the aspect of Wording.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Wording Practice 5 (Enhancing): Consistent on the wording of domain-specific terms.

Example (without this practice):

```
…
LOG.info("Incident ID {}: a new incident is reported.", incID);
…
LOG.info("IncID {}: the incident is closed.", incID);
…
```

Note: Incident ID and IncID refer to the same thing. If possible, developers can consider keeping a consistent convention on the wording of domain terms to mitigate potential confusion.

Q17: This practice can improve the readability of log messages from the aspect of Wording.

A. Strongly agree B. Agree C. Neutral D. Disagree E. Strongly Disagree F. Not sure

Answer:

Comment:

Part 4: Investigate the Readability of Real-world Log Messages

In this part, we randomly select 7 logging statements from real-world open source systems with their surrounding code snippets. The logging statements for each participant are different. For each logging statement that is highlighted with a red rectangle, please examine the readability of its log message based on the aspects of Structure, Information, and Wording. Please only consider the readability of log messages (i.e., the green sentence).

Logging Statement 1

```
try {
    cnxn.sendResponse(hdr, rsp, "response");
    if (request.type == OpCode.closeSession) {
        cnxn.sendCloseSession();
    }
} catch (IOException e) {
    LOG.error("FIXMSG",e);
}
```

Q18-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q18-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q18-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 2

```
    Key key = keystore.getKey(alias, (char[]) keyPasswords.get(alias));
    if (key instanceof PrivateKey) {
        return (PrivateKey) key;
    }
} catch (KeyStoreException e) {
    logger.error("Unable to read private key from keystore", e);
```

Q19-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q19-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q19-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 3

```
try {
  checkOpen();
} catch (IOException e) {
  if (request.hasScannerId()) {
    String scannerName = Long.toString(request.getScannerId());
    if (LOG.isDebugEnabled()) {
      LOG.debug(
        "Server shutting down and client tried to access missing scanner " + scannerName);
    }
    if (regionServer.leases != null) {
```

Q20-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q20-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q20-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 4

```
transport.setMessageListener(this);
connectionManager.addListener(this);
transport.start();
if (transport.boundAddress() != null && logger.isInfoEnabled()) {
    logger.info("{}", transport.boundAddress());
    for (Map.Entry<String, BoundTransportAddress> entry : transport
```

Q21-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q21-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q21-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 5

```
MergeSpecification spec = new MergeSpecification();
for (SegmentCommitInfo info : segmentInfos) {

    if (shouldUpgrade(info)) {

        // TODO: Use IndexUpgradeMergePolicy instead.  We should be comparing codecs,
        // for now we just assume every minor upgrade has a new format.
        logger.debug("Adding segment {} to be upgraded", info.info.name);
        spec.add(new OneMerge(Collections.singletonList(info)));
    }
```

Q22-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q22-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q22-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 6

```
            if (!gossipedToSeed || liveEndpoints.size() < seeds.size())
                maybeGossipToSeed(message);

            doStatusCheck();
        }
    }
    catch (Exception e)
    {
        JVMStabilityInspector.inspectThrowable(e);
        logger.error("Gossip error", e);
    }
    finally
    {
        taskLock.unlock();
    }
```

Q23-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q23-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q23-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Logging Statement 7

```
String username = auth.getUser();
String realm = auth.getRealm();
String domain = auth.getDomain();
if (log.isDebugEnabled()){
    log.debug("{} > D={} R={} M={}", username, domain, realm, auth.getMechanism());
}
if(Mechanism.KERBEROS.equals(auth.getMechanism())) {
```

Q24-1: How do you consider the readability of this log message in the aspect of Structure?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q24-2: How do you consider the readability of this log message in the aspect of Information?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Q24-3: How do you consider the readability of this log message in the aspect of Wording?
A. It's fine B. It's not good C. Not sure

Answer:

Comment:

Part 5: Other Comments and Feedback

Do you have other comments/ideas/concerns regarding the readability of log messages? If so, please share with us below.

Comment: