Logging Checklist

10 Commandments of Logging “Masterzen”

- ❒ **1. Thou shalt not write log by yourself**
  - *Make use of syslog or similar systems as they handle auto-ration and more.*
- ❒ **2. Thou shalt log at the proper level**
  - ***TRACE** track bugs or code routines (dev only, not to be committed to VCS).*
  - ***DEBUG** activated during troubleshooting for debugging.*
  - ***INFO** user-driven or system specific actions.*
  - ***NOTICE** notable events that are not errors (default level for prod), always add context.*
  - ***WARN** events that could result in error, e.g low disk space, always add context.*
  - ***ERROR** for all errors, always add context.*
  - ***FATAL** signifies the end of a program (exit program), always add context.*
- ❒ **3. Honor thy log category**
  - *The category allows classification of the log message. E.g. my.service.api.<apitoken>*
  - *Log categories are hierarchical to allow inherited behaviours, rules and filters.*
- ❒ **4. Thou shalt write meaningful logs**
  - *Treat logging as if there is no access to the program source-code.*
  - *Log should not depend on previous log for context as it may not be there (async).*
  - *Log message prefix **APP-S-SUB-CODE***
    - ***APP** your application name on 3 letters*
    - ***S** severity on 1 letter (e.g. D: debug, I: info, …)*
    - ***SUB** the sub part of the application this code pertains to*
    - ***CODE** a numeric code specific to the error in question*
- ❒ **5. Thy log shalt be written in English**
  - *English is an internationally recognized language.*
  - *English is a better suited technical language.*
  - *English can be written in ASCII characters (stays the same over UTF-8, Unicode, etc).*
- ❒ **6. Thou shalt log with context**
  - *Treat logging as if there is no access to the program source-code.*
  - *Log messages without context are just noise.*
  - *Ensure that at least the local scope of variables are included in the log context.*
- ❒ **7. Thou shalt log in machine parsable format**
  - Text logs are good for humans but very poor for machines.
  - Use a simple international standard format, like JSON.
  - Simplifies automation processing for alerting/auditing.
  - Log parsers require less processing of messages.
  - Log search engine indexing becomes straightforward.
- ❒ **8. Thou shalt not log too much or too little**
  - *Too much logging generates too much clutter (time consuming to browse).*
  - *Too little logging leads to troubleshooting problems (not enough context).*
  - *Log more than enough in dev, then tighten it up before shipping to prod.*
- ❒ **9. Thou shalt think to the reader**
  - *The whole purpose of logging is so that someone will read it one day.*
  - *Readers deserve consistency, so stick to standards like RFC3339 (datetime).*
  - *Log parsers are readers too, so ensure consistency of log format/dictionary.*
- ❒ **10. Thou shalt not log only for troubleshooting**
  - ***Auditing** management/legal events, describe what users of the system are doing.*
  - ***Profiling** logs are time stamped this allows us to infer performance metrics.*
  - ***Statistics** compute user behaviours, e.g. alert when too many errors detected in a row.*