ANEXD

Alexander Austin
Fred Barnes
CO600: Document Type
14 March 2016

# Go QA Checklist

## GoLang

Go lang has a large page at the following https://golang.org/doc/effective_go.html that specifies how all core methodologies should be used as well as various information regarding a documatenation tool and code standards.

- Avoid repetition break things out into functions where possible and use that.
- Variable and function names should be clear and easy to understand. Making it clear what it is doing or means.
- Use camel case for function and variable names with first letter uppercase for functions and lower for variable names which is same as golang native functions and variables.
- Comments at complicated and long sections of code so it is easily to understand what it does.
- Complier will not compile if there are warnings such as unused variables or imports. This helps keep the code clean.
- Keep code lines sort no longer than 100 characters. If it exceeds break it down into more readable format.
- Separate imports into groups with a space e.g.

```
import (
    "fmt"
    "io"
    "log"

    "golang.org/x/net/websocket"
)
```

- Shorter method/var names is better e.g. MarshelIndent over MarshelWithIndentation
- Avoid concurrency in API
- Use godoc tool to turn comments into html pages. Put good well written comments at top of function without any proceeding newlines http://blog.golang.org/godoc-documenting-go-code Example:

```
// Fprint formats using the default formats for its operands and
writes to w.
func Fprint(w io.Writer, a ...interface{}) (n int, err error) {
```

- Use go routinues to manage state
- Avoid go routinues leaks with buffered chans

- Avoid go routinues leaks with quit chan