

# Test BC-LLR-52

"No estado MAINT\_WAIT, caso haja algum erro ao criar o arquivo .LUI, o software deve ir para o estado de ERROR e parar a execução"

## Procedimento realizado

### Análise de código

1. Abrir o código fonte em [modulo\\_bc/components/tftp/tftp.c](#)

## Resultados obtidos

### Código contendo o tratamento de erro ao criar o .LUI

```
modulo_bc > components > tftp > C tftp.c > handle_rrq(int sock, struct sockaddr_in *client, char *filename)
 57 void handle_rrq(int sock, struct sockaddr_in *client, char *filename)
 58     if (bind(transfer_sock, (struct sockaddr *)&transfer_addr, sizeof(transfer_addr)) < 0)
 59         close(transfer_sock);
 60         return;
 61     }
 62     socklen_t addr_len = sizeof(transfer_addr);
 63     getsockname(transfer_sock, (struct sockaddr *)&transfer_addr, &addr_len);
 64     ESP_LOGI(TAG, "Socket de transferência criado na porta %d (TID)", ntohs(transfer_addr.sin_port)); /*BC-
 65
 66     /* BC-LLR-24 Criação do arquivo .LUI
 67     No estado MAINT_WAIT, após receber a requisição de leitura do arquivo de inicialização deve criar
 68     e enviar por um buffer o Load Upload Initialization(LUI) aceitando a operação caso não haja nenhum
 69     problema */
 70     lui_data_t lui;
 71     if (init_lui(&lui, ARINC_STATUS_OP_ACCEPTED_NOT_STARTED, "Operation Accepted") != 0)
 72     {
 73         /*BC-LLR-52 Erro ao criar o .LUI
 74         No estado MAINT_WAIT, caso haja algum erro ao criar o arquivo .LUI,
 75         o software deve ir para o estado de ERROR e parar a execução
 76         */
 77         ESP_LOGE(TAG, "Falha ao inicializar LUI");
 78         close(transfer_sock);
 79         return;
 80     }
 81 }
```