

Chapter 5

Experience

Since we have a well defined test framework and our test cases are testing specific parts of the code, defining the fault injections was straightforward. We just went through our test cases and found the most easily manipulated methods that could be isolated to affect only one or two test cases at a time. This made the overall task better and we completed deliverable 5 along with deliverable 4.

Faults Injection

When defining the fail injections, we tried to inject fails in different functions in order to verify our understanding of all tested functions. To make easier replicating the faults, we added the faults injection code in the functions as comments. Thus, when testing it, one only need to comment the line indicated in the table below, and uncomment the line right below that.

Fault ID	Module	Function	Changes in code	At line	Effects
1	credential.c	credential_match	Remove test for host. CHECK(host)	31	The test 002 shall fail
2	url.c	is_urlschemechar	Change the check for alphanumeric character to only letters. isalnum(ch) -> isalpha(ch)	14	The tests 010 and 011 shall fail
3	url.c	is_url	Remove test for first character. !is_urlschemechar(1, *url++)	23	The test 013 shall fail
4	bisect.c	estimate_bisect_steps	Remove conditional and only returns n.	1032	The test 016 shall fail
5	color.c	git_config_colorbool	As the default, returns the color red instead of the automatic color. GIT_COLOR_AUTO -> GIT_COLOR_RED	282	The test 022 shall fail