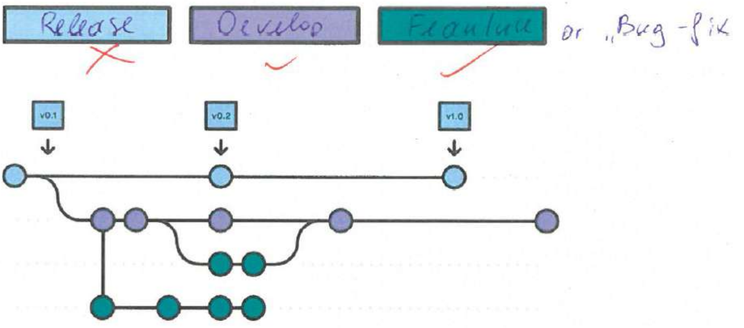
1. (a) Erklären Sie die Programmiersünde *Format String Problems (Bug)*. Bei welcher Art von Befehlen tritt diese auf?
   (b) Geben Sie ein Beispiel in Codeform für einen Format String Bug an.
   (c) Welche Konsequenzen kann ein Format String Bug haben? Nennen Sie eine mögliche Gegenmaßnahme.

2. Betrachten Sie den folgenden Source Code:

```
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
        char input[50];
        int input_lgth = strlen(argv[1]);
        int copy_lgth;
        if(input_lgth >=50)
        {
                copy_lgth = 50;
        }
        else
        {
                copy_lgth = input_lgth;
        }
        strncpy(input, argv[1], copy_lgth);
```

3. (a) Erklären Sie im Kontext von Git die Begriffe *Branches* und *Merge*.
   (b) Im folgenden sehen Sie einen Ausschnitt aus einer mit git-flow erstellten git History. Ordnen Sie den Farben den zugehörigen Branch Typ zu.



   (c) Wie wird die Integrity in einem Git-Repository gewährleistet?

4. Die folgenden Fragen zielen auf das Security-Modell in Webbrowsern ab.
   (a) Nehmen wir an, in einem Webbrowser sind zwei Tabs offen. In einem steht in der Adressleiste `https://www.example.com:443/dir1/index.html`, und in der Adressleiste des anderen Tabs steht `https://example.com/dir2/other.html`. Werden die beiden Seiten als derselbe *Origin* betrachtet?
   (b) Beschreiben Sie den Unterschied zwischen impliziter und expliziter Authentifizierung bei Webapplikationen.
   (c) Was bewirkt die Same-Origin Policy in Webbrowsern?
      (a) Dass beim Aufruf von Ressourcen fremder Seiten die HTTP-Antwort nicht z.B. per JavaScript wörtlich gelesen werden darf.
         ○ Wahr ⊗ Falsch
      (b) Dass innerhalb einer Seite (Origin) clientseitig keinerlei HTTP-Requests zu fremden Seiten gesendet werden können.
         ⊗ Wahr ○ Falsch
      (c) Dass Webseiten keinerlei Ressourcen (JavaScript, CSS, etc.) von fremden Domänen einbinden dürfen.
         ⊗ Wahr ○ Falsch
      (d) Dass *Cross-site Request Forgery*-Angriffe vollständig verhindert werden.
         ○ Wahr ⊗ Falsch

5. (a) Was ist der Unterschied zwischen Authentifizierung und Autorisierung? (3)
   (b) Was können Sie bei der Entwicklung einer Webapplikation gegen das Problem „unsichere direkte Objektreferenzen" tun? Es gibt genau eine richtige Antwort. (2)
      (a) Die Zugehörigkeit des aufgerufenen Objektes zum/zur aktuell angemeldeten Benutzer*In überprüfen.
         ⊗ Wahr ○ Falsch
      (b) Bei jeder Anfrage überprüfen, ob das aktuelle Konto die entsprechende Rolle hat, um die Funktion aufzurufen.
         ○ Wahr ⊗ Falsch
      (c) TLS für die Transportverschlüsselung einsetzen.
         ○ Wahr ⊗ Falsch
      (d) Aufsteigende Objekt-IDs verwenden.
         ○ Wahr ⊗ Falsch

6. (a) Welcher ist der Unterschied zwischen den Maßnahmen gegen Reflected Cross-Site Scripting und Stored Cross-Site Scripting? (3)
      (a) Bei Stored XSS passiert die Ausgabekodierung in der Datenbank, bei Reflected XSS bei der Ausgabe.
         ○ Wahr ⊗ Falsch
      (b) Bei Stored XSS ist die Ausgabekodierung unabhängig vom jeweiligen Ausgabekontext.
         ○ Wahr ⊗ Falsch
      (c) Es gibt keinen grundsätzlichen Unterschied.
         ⊗ Wahr ○ Falsch
      (d) Whitelisting von Eingabeparametern ist bei Stored XSS wirkungslos.
         ○ Wahr ⊗ Falsch

7. (a) Wie funktionieren, generisch gesprochen, Injection-Angriffe, und zwar unabhängig von der Technologie (SQL, STMP, LDAP, etc.)? (3)
   (b) Nehmen wir an, es gebe eine OS-Command-Injection-Lücke in einer Webapplikation, die erfolgreich ausgenützt wird. Im Kontext welches Betriebssystembenutzers werden die injizierten Kommandos *allgemein gesprochen* ausgeführt? (2)
      (a) Mit dem root-Benutzer.
         ○ Wahr ⊗ Falsch
      (b) Als privilegierter Betriebssystembenutzer.
         ○ Wahr ⊗ Falsch
      (c) Als jener Benutzer, under dem der Webserver bzw. Applikationsserver läuft.
         ⊗ Wahr ○ Falsch
      (d) Als nichtprivilegierter Benutzer.
         ○ Wahr ⊗ Falsch