Plano de Gestão de Qualidade
Projeto de Engenharia de Software — Trabalho 2

Caio Delgado João Bronharo Matheus Otenio Vitor Viana

Linguagem: Python **Arquitetura:** MVC **Formato de dados:** CSV

Práticas adotadas para garantir a qualidade

Desde o início, sabíamos que o tempo seria curto e o projeto, mais simples, então buscamos formas práticas de manter a qualidade. Dividimos os módulos com base nas afinidades de cada um: Caio ficou com as configurações e o dashboard, Matheus com o chat em grupo, Vitor com a criação automática de grupos e João com o fórum de perguntas e respostas.

Tentamos aplicar alguns padrões de projeto vistos em aula, como *Singleton*, *Factory Method*, *Observer* e *Command*, mas sempre de maneira simples e direta, para evitar complicações desnecessárias.

Tivemos reuniões no Discord, onde cada um mostrava seu progresso e discutíamos próximos passos. Na prática, as reuniões acabaram acontecendo a cada dois dias. Para dúvidas rápidas, usávamos o WhatsApp. O GitHub serviu para versionamento, mas enfrentamos alguns problemas com ele (comentados a seguir). O Google Drive, por outro lado, funcionou bem por sua simplicidade e serviu como backup do projeto.

Desafios enfrentados relacionados à qualidade

Um dos maiores desafios foi integrar os módulos. A integração só aconteceu quando cada parte estava praticamente finalizada e sem muitas mudanças, o que acabou atrasando um pouco o processo. Inicialmente tentamos fazer as importações diretas entre pacotes, mas isso causou muitos erros de importação cruzada. Assim, foi decidido usar chamadas de `subprocess`, o que resolveu boa parte dos problemas e tornou os módulos mais independentes.

Tivemos também um problema sério no GitHub: um commit acabou sobrescrevendo arquivos importantes do módulo de fórum. Sem ficar no histórico de commitis um dos arquivos acabou sendo perdido, dessa forma, combinamos de utilizar o Google Drive como forma de backup e sempre manter eles atualizados.

Por fim, mudamos o escopo: inicialmente queríamos adicionar notificações entre os módulos, mas vimos que isso exigiria uma integração muito mais complexa. Resolvemos mudar para um fórum, que cumpria a função de interação sem criar tantas dependências entre os sistemas.

Validação da qualidade final do sistema

Cada um testou seu próprio módulo manualmente, mas também fizemos duas reuniões focadas só na validação do sistema de cada membro do grupo. Simulamos o uso por um usuário comum, desde o início do fluxo, e fomos anotando os bugs.

Durante esses testes, por exemplo, percebemos que o chat não carregava corretamente quando era iniciado via `subprocess`. Resolvemos isso alterando a forma de leitura dos dados no CSV.

Apesar de não termos usado ferramentas automatizadas de análise de código, discutimos e revisamos em grupo alguns trechos mais complexos durante as reuniões. Isso ajudou a detectar falhas e melhorar partes do código.

Considerações finais

No fim, conseguimos entregar um sistema funcional. Algumas práticas funcionaram muito bem: o uso do GitHub facilitou bastante a organização, e a divisão clara de tarefas evitou sobrecarga. Esse mesmo, apesar de essencial, acabou gerando problemas por falta de coordenação nos commits.

O que faríamos diferente seria detalhar melhor o formato dos arquivos CSV desde o começo. Isso teria evitado vários erros na hora de integrar os módulos. Também teríamos feito backups em mais de um local — perdemos algumas anotações por confiar demais em um único lugar.

Aprendemos que comunicação constante, mesmo que rápida, é essencial. E que revisar o que o outro está fazendo ajuda a manter a coesão do projeto.