Aqui estão os principais pontos da aula:

1. **Engenharia de Software:**
   * Processo sistemático para criar software de alta qualidade, adaptando-se às mudanças e focando em qualidade e rapidez de entrega.
   * Compreende instruções executáveis, estruturas de dados e documentação relacionada.
2. **Crise do Software (década de 1960):**
   * Desafios como ineficiências, desenvolvimento falho e comunicação deficiente entre clientes e desenvolvedores.
   * Mitigação desses problemas com abordagens metodológicas avançadas.
3. **Modelo de Processo de Software:**
   * Conjunto de práticas e procedimentos que orientam o desenvolvimento, desde a concepção até a entrega e manutenção.
   * Importância de adaptar o modelo às características específicas de cada projeto.
4. **Métodos Ágeis:**
   * Manifesto Ágil de 2001, que enfatiza flexibilidade, colaboração e adaptabilidade.
   * Scrum: Método ágil mais utilizado, organiza o desenvolvimento em sprints e promove auto-organização das equipes.
   * Extreme Programming (XP): Práticas rigorosas para planejamento, projeto, codificação e testes.
5. **Requisitos de Software:**
   * Requisitos funcionais e não funcionais são essenciais para guiar o desenvolvimento.
   * Engenharia de requisitos envolve elicitação, análise, especificação e validação de requisitos.
6. **Controle de Versões:**
   * Gerenciamento da Configuração do Software (GCS) é vital para controle de mudanças e rastreabilidade.
   * Ferramentas como Git e GitHub são amplamente utilizadas para gerenciamento de versões.
7. **Abordagem Híbrida:**
   * Combinação do método cascata (para análise e planejamento iniciais) e Scrum (para desenvolvimento iterativo).
   * Uso de Git para controle de versões e colaboração.
8. **Resultados:**
   * Abordagem híbrida permite flexibilidade e adaptação, garantindo o lançamento dentro do prazo e acomodando mudanças nos requisitos.

4o