

1) existe vários problemas que encontramos em um ciclo de vida, que dificilmente localizamos no início do ciclo de vida, mas que podem ter grande importância no final, então no método de análise visto em sala de aula, percebemos que quando ocorre uma verificação dos requisitos, quando levantamos ver se os requisitos atenderam a demanda do cliente, percebemos que as vezes tomamos focos diferente, pois como comentado em sala o software não é algo muito simples não é um projeto complexo, onde não existe critério de segurança em seu desenvolvimento, mas para atender essas demandas, o software por ser complexo, não pode se prever tudo que vai acontecer nele, as vezes processos que são solicitados não trabalham bem como também o comportamento que eles farão para o cliente não seja exatamente como solicitou, com isso se desencadeia um grande problema, pois para fazer as correções, vai exigir custo, deixando o software muito mais caro do que foi planejado, outro problema muito comum nesse processo, é o cliente solicitar algo que não vai utilizar, uma demanda desnecessária, vindo acarretar custo maior pro cliente e perda no tempo da produção do software também. Um grande problema que ocorre no decorrer do tempo, é o fim de ciclo de vida, onde o software não é mais viável as mudanças que ocorreram durante esse tempo, ficando assim um software obsoleto.

2) Manifesto ágil, é uma forma de construir software onde deixa os métodos comuns para trás, coletando só o que é mais útil dos métodos normais, e mantendo algumas boas práticas de produção de software, mas seu início não foi só uma pessoa que fez e sim 17 líderes de projetos relevantes, se reuniram com esse intuito de melhorar a forma de construção de software, pois as formas comuns já estavam ultrapassadas, se tornando mais caras, e difíceis de se adequar perante a todos os problemas já citados na questão anterior.

3)  
valores:

- 1) os indivíduos e suas interações acima de procedimento e ferramenta;
- 2) funcionamento de software acima de documentação abrangente;
- 3) A colaboração do cliente acima da negociação e contrato;
- 4) A capacidade de resposta de mudanças acima de um plano preestabelecido;

- 4) 1) Garantir a satisfação do cliente, entregando rápida e continuamente software funcional;
- 2) Até mesmo mudanças tardias de escopo no projeto são bem-vindas.
- 3) Software funcional é entregue frequentemente (semanal ou mensal - o menor intervalo possível);
- 4) Cooperação constante entre as pessoas que entendem do 'negócio' e os desenvolvedores;
- 5) Projetos surgem por meio de indivíduos motivados, devendo existir uma relação de confiança.
- 6) A melhor forma de transmissão de informação entre desenvolvedores é através da conversa 'cara a cara'

5)  
1) facilidade de adaptação: 1- entrega de software funcional; 2- mudanças sempre bem vindas; 3- excelência técnica e bom designer

2) pessoas e interações: 1- negócios e desenvolvedores trabalham próximo; 2- Conversas cara a cara; 3- time auto ajustável; 4- promoção de desenvolvimento sustentável; 5- indivíduos motivados;

- 6) 1- arquitetura tem que ser mais complexa: Toda mudanças dever se ter cautela.  
2- controle sobre mudanças e maior: as mudanças ocorrem, e assim fica sem uma linha onde pode se guiar, desconfortável deste modo;  
3- requer uma grande participação do cliente: se tornando centro do processo de desenvolvimento, assim surgindo o problema maior que seria a disponibilidade do cliente para estar presente.