

MÓDULO 3 – Métodos Ágeis

Questionário¹

Sobre o Manifesto Ágil

1. Como a “colaboração do cliente” pode ser feita ao longo de projetos de software? (3º valor)
2. Se mudanças são bem-vindas ao longo do projeto, como fazer com o orçamento? (2º princípio)
3. Como e quem pode fornecer o ambiente e suporte necessário para o desenvolvimento de um software? Avalie esse questionamento no contexto da sociedade atual. (5º princípio)
4. O que vem a ser “vantagem competitiva para o cliente”? (2º princípio)
5. Quem são as “pessoas de negócio”? (4º princípio)
6. Como balancear entrega contínua e aceitação de mudanças? (1º princípio)
7. O ritmo constante não pode comprometer a flexibilidade e *vice-versa*? Discuta. (8º princípio)
8. O que vem a ser uma equipe “auto-organizada”? É viável prescindir de gerentes de desenvolvimento? (11º princípio)
9. Como uma equipe ágil pode implementar o “desenvolvimento sustentável”? É possível garantir o ritmo constante em qualquer situação? (8º princípio)
10. O investimento em excelência técnica e bom design não podem comprometer a agilidade e, conseqüentemente, as entregas rápidas? (9º princípio)
11. A falta de documentação abrangente não pode afetar a manutenção e evolução do software? Como definir os limites dessa abrangência? (2º valor)
12. Simplicidade em excesso não pode reduzir o valor e a qualidade do software? (10º princípio)

Questões sobre o SCRUM

1. Quais são os elementos que caracterizam a auto-organização de uma equipe Scrum?
2. A prioridade das histórias de usuário devem ser definida também em função da complexidade para o desenvolvimento? Explique.
3. Considerando que os eventuais *bugs* encontrados ao longo dos *sprints* em um projeto desenvolvido com o *Scrum* podem ser densos a ponto de se tornarem novas tarefas, qual o impacto de adiar a execução dessas tarefas para futuras iterações? Discuta.

1 Os itens deste questionário nasceram em atividades anteriores da disciplina Engenharia de Software I. Pedia-se aos alunos que, reunidos em grupos, formulassem questões a serem propostas para outros grupos em dinâmica desenvolvida em sala de aula.

4. A realização do *daily scrum* pode comprometer o desempenho de times experientes? Explique.
5. Por que métodos como o *Scrum* são mais adequados do que os processos prescritivos (ex. RUP) para projetos desenvolvidos em cenários considerados caóticos (usando a qualificação de um dos vídeos utilizados), isto é, em situações em que há muitas incertezas?
6. Considerando a diferença de produtividade entre os membros que formam uma equipe *Scrum*, como é possível realizar boas estimativas de tempo para a conclusão de um conjunto de histórias de usuário?
7. Como o *Scrum* lida com problemas encontrados ao longo dos *Sprints*, visto que são sujeitos a um *timebox* que não pode ser alongado?
8. Em que momentos do cerimonial proposto pelo *Scrum* a colaboração do cliente é mais relevante? Explique.
9. Explique a relação existente entre *sprint* e *release*?
10. O *scrum master* é o gerente do projeto *Scrum*? Explique.
11. Qual a estrutura do *burndown chart* e como pode ser utilizado para auxiliar o gerenciamento do projeto ao longo de um *sprint*?
12. Como se identifica a queda ou o aumento de desempenho no *burndown chart*? O que se pode interpretar quando há redução contínua de desempenho ao longo do projeto? E se ocorrer o contrário, com desempenho sempre acima do esperado?

Questões sobre Extreme Programming

1. No XP, a presença constante do cliente e a possível interferência com mudanças de requisitos não traz risco ao cronograma estabelecido e, assim, ao cumprimento das atividades planejadas para uma iteração? Explique.
2. Em projetos de larga escala desenvolvidos com o XP por uma equipe de desenvolvimento numerosa, como é possível conciliar o desenvolvimento e os testes em *timebox* muito curtos?
3. Como lidar com os riscos relativos ao comprometimento da integridade da codificação do produto em relação às práticas “posse coletiva” e “integração contínua”?
4. Que vantagem pode trazer o desenvolvimento em pares (*pair programming*), visto que cada desenvolvedor poderia trabalhar simultaneamente para implementar diferentes histórias de usuário?
5. Em que aspectos o uso da metáfora pode ser útil a um projeto XP?
6. Qual a importância do TDD para a integração contínua?
7. Em que aspectos o XP facilita ou dificulta a integração de novos membros de equipe em um projeto em andamento?

8. Como garantir a consistência do produto em desenvolvimento incremental visto que, no XP, qualquer membro da equipe pode refatorar código já integrado a qualquer momento?