

Perguntas do slide Processo unificado – parte 1

1) Na sua opinião, por que projetos que utilizam o modelo em cascata falham mais do que os que utilizam o modelo iterativo e incremental?

R: Porque o fato de uma fase ter de ser muito bem esclarecida para somente então prosseguir-se para outra é uma filosofia que não condiz com o desenvolvimento de software atualmente. Primeiramente, ao procurar uma empresa que desenvolva sistemas, até mesmo o próprio cliente tem dúvidas sobre o que ele realmente quer dos desenvolvedores e do produto final que ele deseja, portanto é quase que impossível definir com certeza quais serão todas as funcionalidades e requisitos desse sistema (definição essa que é imposta pelo método *waterfall*). Além disso, a falta de um protótipo também requer demais do *stakeholder*, que só será capaz de testar o produto quando todo ele estiver concluído; às vezes, o resultado não é o que o cliente esperava, e isso pode ser muito custoso para aqueles que produziram o software e frustrante para quem requisitou esse produto. Para essas situações (que englobam a maioria dos sistemas produzidos na atualidade), a filosofia do modelo incremental supre esses defeitos, pois engloba as mesmas fases existentes no modelo cascata e sua progressão, porém, ao invés de produzir o software inteiro, pequenas funcionalidades dele vão sendo construídas em cada um dos incrementos (daí o nome). Isso significa, por exemplo, que mesmo exigindo um conjunto de requisitos bem definidos durante a fase de análise, essas funcionalidades bem explanadas irão condizer somente com uma porção do software, e não ele todo. Ao final de cada incremento, o cliente é capaz não só de ver o que está funcionando como também testá-lo por si próprio, inferindo conclusões acerca dessa funcionalidade e permitindo o refinamento dos requisitos que compõem o restante do produto, gerando um resultado final mais próximo aos desejos do *stakeholder*.

2) Discorra sobre as principais características do PU apresentando suas vantagens e desvantagens (se for o caso).

R: O processo unificado agrega o modelo incremental explanado acima mais o paradigma de orientação a objetos, tendo como ferramenta de apoio a UML (ou *universal modeling language*), amplamente utilizada para definir projetos no padrão previamente mencionado. Esse processo é um poderoso *framework* que, dado sua natureza, precisa ser ajustado de acordo com as necessidades do projeto em questão – utiliza-se dele somente o que se precisa, dado a ampla quantidade de recursos disponíveis. A metodologia possui, em sua essência, algumas fortes características: O modelo incremental é utilizado aqui, dividindo sua funcionalidade em quatro fases (concepção, elaboração, construção e transição) que são regidas por disciplinas (requisitos, análise e projeto, entre outras), que produzem um incremento no fim. O PU é dirigido por casos de uso, que especificam qual o escopo do sistema em questão e o papel de cada um dos atores no relacionamento com o software – é por meio desses que se identificam quais requisitos são os mais complexos, começando as iterações a partir deles. O modelo também é centrado arquiteturalmente, o que infere o comprometimento a um modelo conceitual – descrevendo pacotes, classes e interfaces - que servirá de alicerce para a construção de todas as funcionalidades do produto. Uma das últimas características do processo diz respeito ao foco em riscos: Durante a especificação dos incrementos, os que representam um grande risco ao software (risco esse que é identificado, analisado e classificado durante a fase de concepção) serão os primeiros a serem implementados, visando tomar um plano de contingência dos riscos ao produto final. A principal vantagem do processo é a união de uma filosofia incremental (que na atualidade é quase que obrigatória para a construção de sistemas) com a orientação a objetos, paradigma que ganhou sua fama por ser reutilizável, fácil de realizar manutenção e poder ser estendido facilmente. Outra vantagem também se dá no fato de que outros métodos de desenvolvimento de software podem ser acoplados a esse: juntando a filosofia ágil ao processo unificado, temos tudo explanado acima como também

incrementos feitos em curtos períodos de tempo e pouca documentação produzida. O fato do cliente estar em constante contato com a equipe de desenvolvimento, podendo testar funcionalidades simples e redefinir especificações contribui muito para um produto final satisfatório. Entretanto, alguns problemas podem ser destacados na escolha dessa metodologia: a utilização de casos de uso não define por completo os requisitos do cliente, e a participação desse último na construção desses modelos é quase que nula. Em projetos grandes com utilização de tecnologia de ponta, o conceito de reusabilidade da orientação a objetos se perde, o que torna a economia de tempo impossível de acontecer. Apesar de seus defeitos, o PU é bastante completo por si só, e aborda os pontos essenciais de um projeto (riscos e arquitetura, por exemplo) de maneira direta.