Artigo – How Microsoft Builds Software

* Ciclo de Vida - Sincroniza e Estabiliza. Há diversos paralelos com XP.
* Fase de planejamento

Os gerentes de produto (especialistas em marketing) criam uma declaração de visão: metas para o novo produto, identificadas através das necessidades dos clientes.

Baseados na visão, gerentes de programa, em conjunto com desenvolvedores, escrevem um documento de especificação delineando funcionalidades em profundidade suficiente para organizar calendário e alocar pessoal. O documento não tenta capturar todos os detalhes.

Baseado no documento de especificação, a agenda é coordenada e são organizados times de features, com 1 gerente de programa, 3 a 8 desenvolvedores e 3 a 8 testadores, 1 para 1.

* Fase de desenvolvimento

Os gerentes de programa coordenam evolução da especificação. Desenvolvedores projetam, codificam, prototipam e debugam. Produto e testes são desenvolvidos em paralelo. Testes contínuos. Desenvolvedores e *test buddies* trabalham juntos, lado a lado*.* São encorajados a experimentar e mudar.

São pequenos times, trabalhando em paralelo, organizados ao redor de *features.* As *features* são priorizadas e distribuídas em 3 ou 4 subprojetos (marcos). Cada time realiza ciclos completos. As sincronizações são diárias ou semanais.

As features mais críticas e os componentes compartilhados são desenvolvidos primeiro. Baseados nas necessidades dos clientes. Em seguida as demais funcionalidades, nas outras fases.

* Fase de estabilização

Estabilização em incrementos, através da correção de erros e sincronização. Continuamente encontrando e corrigindo erros. Permitem aos times verificar o quanto do produto foi desenvolvido. Uma data de lançamento fixa e múltiplos ciclos de lançamentos.

Feedback e monitoramento contínuo do consumidor, através de laboratórios. Desenvolvedores realizam debug final e estabilização do código. Testadores recriam e isolam erros.

Regras: commit do código deve ser feito em um horário fixo. Cria-se uma nova buid após esse horário. Caso o desenvolvedor commite um código que gera defeito na build, deve corrigi-lo imediatamente.

Há testes internos do produto completo, testes externos através de beta e preparação do release candidate.

Design de produto e processo de forma a que times grandes trabalhem como times pequenos. Facilita a coordenação e comunicação. Estratégia de entregar produtos “bons o suficiente”, não “perfeitos”. Utilização de métricas para aferir os milestones.

Desvantagens – há necessidade, em certos produtos, de modelos matemáticos, métodos formais e uma sequência tradicional de desenvolvimento.

Vantagens – divide grandes produtos em pedaços gerenciáveis; “Sempre” há um produto pronto para a venda; adequado para mercados de mudança rápida, com ciclos de vida curtos; teste exaustivos.

* Semelhante a XP: exceto design simples e propriedade coletiva do código. Ssuperficialmente á sobreposição de 80%. Útil pois nem todas as especificações são evidentes/existem no início do projeto.