

UTILIZAÇÃO DE FRAMEWORKS NO DESENVOLVIMENTO DE SISTEMAS WEB

1º SGT ENG MARCOS PAULO MIRANDA DE SOUZA
Graduado em Sistemas de Informação

RESUMO. O GRANDE DESAFIO NA CONSTRUÇÃO DE SISTEMAS, LEVAM A CONSTANTE EVOLUÇÃO DA ENGENHARIA DE SOFTWARES, QUE TENTA DIMENSIONAR QUAIS AS BOAS PRÁTICAS A SEREM SEGUIDAS PARA QUE O PRODUTO FINAL SEJA ALGO CAPAZ DE EVOLUIR, ATENDER AS NECESSIDADES EXISTENTES E SOLUCIONAR OS PROBLEMAS PROPOSTOS INICIALMENTE. NESSE PROCESSO DE EVOLUÇÃO SURGEM FATORES COMO O GERENCIAMENTO DO TEMPO, CUSTOS, CURVA DE APRENDIZAGEM PARA A UTILIZAÇÃO DE FERRAMENTAS E QUAIS TECNOLOGIAS UTILIZAR. É NESSE PREÂMBULO QUE ENCONTRA-SE A UTILIZAÇÃO DE FRAMEWORKS COMO UMA SOLUÇÃO MUITO ADOTADA QUE ATINGE TODOS ESSES PONTOS COM ÊXITO, PROPORCIONANDO GANHOS QUE POSSIBILITAM MAIOR FLEXIBILIDADE, AGILIDADE, ESTABILIDADE, SEGURANÇA, PADRONIZAÇÃO DE TECNOLOGIAS E MANUTENÇÃO MAIS SUSTENTÁVEL. VÁRIAS FERRAMENTAS DESSA ÁREA SÃO APRESENTADAS TODOS OS DIAS, COM GRANDES POSSIBILIDADES DE UTILIZAÇÃO E TECNOLOGIAS DIFERENTES, SENDO UTILIZADAS SEM UMA ANÁLISE A LONGO PRAZO SOBRE A SUA CAPACIDADE DE SUPORTE ÀS CONSTANTES EVOLUÇÕES DOS SISTEMAS QUE OPERAM VIA INTERNET (SISTEMAS WEB). DESSA FORMA, SERÃO APRESENTADOS ALGUNS CRITÉRIOS QUE DEVEM SER LEVADOS EM CONSIDERAÇÃO ANTES DE UMA EMPRESA OU UM PROGRAMADOR SELECIONAR UM FRAMEWORK PARA USO.

PALAVRAS-CHAVE: FRAMEWORK. PROGRAMAÇÃO. DESENVOLVIMENTO. ENGENHARIA DE SOFTWARES.

INTRODUÇÃO

Dentre as possibilidades de escolha de frameworks, o programador PHP se depara com uma gama de opções a seu dispor. A título de exemplificação o mercado oferece o CakePHP, Symfony, Zend Framework, CodeIgniter, Yii 2, Phalcon, Prado, entre outros.

Deve-se levar em consideração, para escolha de uma ferramenta, os fatores que foram mais explorados em cada um dos frameworks e que proporcionaram vantagens em alguns aspectos, entendendo que cada um tem o seus pontos fortes e fracos, para cada tipo de projeto e objetivo.

Os fatores mais importantes que devem ser considerados num framework são estes: estabilidade, segurança, performance, curva de aprendizado, recursos técnicos disponíveis e flexibilidade.

Segundo Alvim (2010, p.12),

O framework é um conjunto de classes que colaboram entre si, proporcionando melhores práticas de desenvolvimento e diminuição à repetição de tarefas. Além disso, evita variações de 'soluções diferentes

para um mesmo tipo de problema'. O que facilita a reutilização e customização dos códigos.

A utilização dessas ferramentas tem sido largamente aplicada em grandes projetos e possuem, armazenadas dentro de suas estruturas, vários princípios da engenharia de softwares, que tem sido estudados até hoje.

1 DESENVOLVIMENTO

1.1 ESTABILIDADE

A estabilidade pode ser vista sobre dois aspectos principais: a capacidade do framework de aderir-se às novas tecnologias sem a ocorrência de erros e a capacidade de sofrer updates em relação a funcionalidades já existentes sem que ocorram erros.

Fagan (1986) relatou que mais de 60% dos erros em um programa podem ser detectados por meio de inspeções de programas. No processo *Cleanroom* (PROWELL, 1999), afirma-se que mais de 90% dos defeitos podem ser descobertos em inspeções de programas.

Quanto às inovações tecnológicas que



ocorrem constantemente e alteram o mercado de negócios, é certo afirmar que tais mudanças exigem adaptações dos recursos existentes nos frameworks para melhor atender a esse mercado. Os rumos são ditados pelas grandes empresas que escolhem uma determinada área da tecnologia a ser explorada em busca do lucro financeiro. Os analistas e programadores se deparam com a necessidade emergencial de fazer com que o seu sistema possa atingir determinada camada de clientes ou um determinado nicho deste e isso exige que o framework, que foi selecionado, seja capaz de produzir artefatos que possam ser utilizados nessa nova direção. O problema está no fato de isto não depender apenas das habilidades do programador, mas talvez de uma melhoria do framework escolhido. Esse trabalho é feito pela equipe de programadores, que são os autores da ferramenta. É nesse momento, que se identifica quais tecnologias irão sobreviver as mudanças do mercado e quais se tornarão obsoletas.

As atualizações dos frameworks não podem causar grandes impactos nos sistemas que já foram desenvolvidos para sua utilização, sob o risco de tornar inviável a aplicação dessas atualizações em projetos grandes que já foram implementados.

A nova versão do framework deve apresentar uma solução com o menor impacto possível e sem produção de erros, atingindo a performance de uma ferramenta estável.

1.2 SEGURANÇA

A segurança é uma das características mais importantes de um sistema, mas que somente é valorizada sua invasão. Normalmente, o problema remonta ao projeto estrutural, no qual os investimentos relacionados à segurança compõe parte ínfima do total investido no projeto.

O termo 'confiança' foi proposto por Laprie (1995) para cobrir os sistemas relacionados com atributos de disponibilidade, confiabilidade, segurança e proteção.

Um sistema deve ter, na sua lista de requisitos, os caso de segurança a serem abordados e atendidos desde a fase inicial da construção do sistema. De uma forma mais concisa, Bishop e Bloomfield (1998) definem um caso de segurança como:

Um corpo de evidências documentado, que fornece argumentos convincentes e válidos de que um sistema é suficientemente seguro para determinada aplicação, em determinado ambiente.

Um dos erros mais comuns, na aplicação de preceitos da engenharia de softwares, é a preocupação com requisitos de segurança, somente depois da fase final de desenvolvimento. Tais requisitos devem ser planejados e previstos desde a fase da Análise de Requisitos, sendo, constantemente, monitoradas e aperfeiçoadas durante cada implementação.

A área de segurança de sistemas é muito vasta e, para uma única equipe de desenvolvimento abranger o estudo de tantas formas de ataques possíveis e vulnerabilidades descobertas a cada dia, seria realmente muito dispendioso e trabalhoso, interferindo em questões de custo final e prazos.

Dessa forma, a utilização de um framework amenizaria essa preocupação porque já tem funcionalidades voltadas para a segurança no seu escopo. Além disso, contribui para a diminuição dos gastos financeiros em pesquisas e estudos sobre segurança e de outros fatores incluídos como tempo de desenvolvimento. Portanto, ter em mãos uma ferramenta que já atende em sua estrutura aos requisitos de segurança básicos, embutidos em sua tecnologia, limita às equipes de desenvolvimento apenas o trabalho específico do tratamento de segurança do sistema.

1.3 PERFORMANCE

A performance é percebida de maneira mais abrupta, quando o framework é utilizado ao ponto de, quase, esgotar os seus recursos de hardware, podendo sobrecarregar a estrutura que o mantém, tornando inviável a sua uti-



lização.

Os requisitos de performance, quando não atendidos da maneira ideal, podem ser compensados através da utilização de mais recursos de hardware, mas essa é uma maneira mais custosa, financeiramente, para a resolução desse problema. O ideal na criação de um sistema é obter um equilíbrio entre performance e a quantidade de recursos ativos, que também são requisitos do sistema. Com esse entendimento, não podemos ter um sistema que atenda a tantos requisitos de segurança, que causem baixa performance no desempenho de funcionalidades básicas.

Essa avaliação, do equilíbrio entre a performance e os recursos concorrentes do sistema, deve ser explorada pela equipe de desenvolvimento para definir o framework a utilizar.

1.4 CURVA DE APRENDIZADO

No momento da adoção de novas tecnologias no desenvolvimento de um sistema, deve ser avaliado o tempo que leva para ter uma equipe de desenvolvimento plenamente adaptada àquela nova tecnologia. E entre os recursos necessários, o custo para treinamento dessa equipe. O gerente da equipe de desenvolvimento tem uma importante decisão: qual tecnologia deve ser utilizada no desenvolvimento para atender aos requisitos e os detalhes técnicos apontados pelo arquiteto de sistemas, cumprindo os prazos exigidos e tendo por base a curva de aprendizado registrada por equipes anteriores, que mostram o tempo médio para a obtenção de expertise num determinado framework.

Em alguns casos, mesmo que o tempo de aprendizado seja longo em relação a um determinado framework, há ganhos no desenvolvimento de outros sistemas requisitados que utilizam a mesma tecnologia. Tudo isso depende da demanda da equipe em relação ao framework, sabendo que a principal finalidade da adoção de qualquer uma dessas ferramentas está, justamente, na velocidade de produ-

ção de artefatos de sistemas.

1.5 RECURSOS TÉCNICOS DISPONÍVEIS

Os recursos técnicos disponíveis dependem da equipe de desenvolvimento, da empresa ou instituição a que pertencem. Isso também pode ser chamado de escalabilidade, visto que um sistema, na maioria das vezes, aumenta de tamanho devido as novas necessidades que aparecem durante o seu ciclo de vida.

A escalabilidade de um sistema reflete sua capacidade de oferecer um serviço de alta qualidade, uma vez que aumenta a demanda de sistema. Neuman (1994) identifica três dimensões da escalabilidade:

1. **tamanho.** Deve ser possível adicionar mais recursos a um sistema para lidar com um número crescente de usuários;
2. **distribuição.** Deve ser possível dispersar geograficamente os componentes de um sistema, sem comprometer seu desempenho;
3. **capacidade de gerenciamento.** É possível gerenciar um sistema à medida que ele aumenta de tamanho, mesmo que partes dele estejam localizadas em organizações independentes.

A equipe de desenvolvimento deve possuir pessoas com conhecimentos técnicos variados. É desejável que seus integrantes tenham qualificações técnicas que se complementem entre si, concomitantemente, com a experiência pessoal adquirida ao longo da carreira. Somente nesse caso, não haverá um grande investimento inicial no treinamento da equipe, haja vista a complementariedade de suas expertises, experiência pregressa.

Na análise da forma de trabalho da empresa, influenciando diretamente no aperfeiçoamento dos seus funcionários, podemos encontrar uma diretriz que prima pelo aperfeiçoamento de suas equipes através de treinamentos dentro do próprio local de trabalho. Essas são formas mais viáveis, economicamente e funcionalmente, para algumas empresas.



Com isso, a empresa deixa de iniciar a busca por um funcionário mais qualificado, evitando perda de tempo na adaptação e integração com a equipe e projetos em andamento.

1.6 FLEXIBILIDADE

A flexibilidade reflete a capacidade de um framework se adaptar mais facilmente as tecnologias existentes, sem a obrigatoriedade de mudar as formas de abordagem em relação a segurança, implementação e performance. Por exemplo, um framework robusto deve oferecer a possibilidade de trocar o tipo de banco de dados do sistema, como funcionalidade nativa da ferramenta. Um framework dessa categoria poderia mudar de MySQL para Postgres, sem fazer modificações internas além da mudança de poucos parâmetros na chamada de um objeto a ser instanciado.

A adoção de uma ferramenta com as características supracitadas redundam em ganho de tempo e custo no desenvolvimento de um sistema.

CONCLUSÃO

É inegável que a utilização de frameworks tem sido uma ferramenta muito útil na aplicação de princípios da engenharia de softwares, devido à constatação prática de ganho no tempo de desenvolvimento, economia de recursos, capacidade de adaptação, atendimento a requisitos de segurança e performance. Porém, todas essas características embutidas numa única ferramenta não descartam a interferência direta de uma equipe de desenvolvimento nos requisitos do sistema, uma vez que tais ferramentas apenas fornecem a estrutura básica que serve de alicerce às demais implementações. Essas ferramentas não devem desestimular a criatividade para a criação de novas funcionalidades em frameworks.

A utilização e o surgimento de novos frameworks têm contribuído de forma bastante positiva no desenvolvimento de programas e

sistemas, mudando não só a perspectiva de agilidade dos desenvolvedores, mas também, a facilidade de acesso a novas tecnologias. A utilização desse tipo de ferramenta tornou-se essencial no mundo de hoje, desmistificando um pouco o ofício do programador e facilitando o acesso ao desenvolvimento de novas tecnologias.

USE OF FRAMEWORKS IN THE DEVELOPMENT OF WEB SYSTEMS

ABSTRACT. THERE ARE NOW A RANGE OF OPTIONS FOR THE PHP PROGRAMMER, FOR EXAMPLE, CHOOSE BETWEEN FRAMEWORK AND AMONG THEM WE HAVE: CAKEPHP, SYMFONY, ZEND FRAMEWORK, CODEIGNITER, YII 2, PHALCON, PRADO, AMONG OTHERS. WE MUST TAKE INTO ACCOUNT, AT THE TIME OF CHOOSING A TOOL, THE FACTORS THAT WERE MOST EXPLOITED IN EACH OF THE MOST USED FRAMEWORKS AND THAT PROVIDED ADVANTAGES IN SOME ASPECTS, UNDERSTANDING THAT EACH HAS ITS STRENGTHS AND WEAKNESSES, DEPENDING ON THE TYPE OF PROJECT AND ITS PURPOSE. THE MOST IMPORTANT FACTORS THAT SHOULD BE CONSIDERED IN A FRAMEWORK ARE ITS STABILITY, SECURITY, PERFORMANCE, LEARNING CURVE, AVAILABLE TECHNICAL RESOURCES AND FLEXIBILITY. THE USE OF THESE TOOLS HAS BEEN WIDELY APPLIED IN LARGE PROJECTS AND HAS STORED WITHIN ITS STRUCTURES SEVERAL PRINCIPLES OF SOFTWARE ENGINEERING THAT HAVE BEEN STUDIED UNTIL TODAY.

KEYWORDS: FRAMEWORK. DEVELOPMENT. PROGRAMMING. SOFTWARE ENGINEERING.

REFERÊNCIAS

ALVIM, Paulo. Tirando o Máximo do Java EE 6 Open Source com jCompany® Developer Suite. 3. Ed. Belo Horizonte: Powerlogic Publishing, 2010. 12p.

Sommeville, Ian. Engenharia de Softwares, 9 Ed. São Paulo: Perarson Prentice Hall, 2011.

O autor é graduado em Sistemas de Informação, possui interesse em programação web, Linux e Banco de dados MySQL. Atualmente, exerce a função de monitor da Escola de Comunicações e pode ser contactado pelo email sgtmarcos@yahoo.com.br.

