

Manifesto Hotmart V2.0.0

Vitor Lopes
vitor.lopes@hotmart.com

12 de dezembro de 2021

Resumo

A Hotmart é uma excelente empresa para se frequentar, se fazer amigos, além da interação e da troca de conhecimento. Essas características são tão fortes que estão presentes em seus pilares e mantras descritos em seu código de conduto. Entretanto, nem tudo são flores, principalmente no que se diz respeito às regras de negócio que ficam por vezes ocultas nos códigos ou na mente de quem desenvolveu, além de existirem processos e sistemas legados. Este material tem como objetivo propor uma metodologia ágil e divertida tentarmos lidar com esses desafios.

1 Introdução

Eis algumas definições encontradas pela internet que descrevem o Software Legado¹:

1. dificuldade de compreensão das regras de negócio neles implementadas;
2. desconhecimento das razões que levaram a determinadas decisões;
3. problemas na estruturação dos módulos de código;
4. miscelânea de estilos de programação;
5. obsolescência das ferramentas de desenvolvimento;
6. impossibilidade de reaproveitamento dos equipamentos nos quais são executados para execução de softwares mais atuais;
7. Altos custos de manutenção;
8. Software complexo;
9. Software de suporte obsoleto;
10. Hardware obsoleto;
11. Sem conhecimento técnico;
12. Negócio crítico;
13. Backlog de solicitações de mudança;

¹https://pt.wikipedia.org/wiki/Sistema_legado

- 14. Documentação deficiente;
- 15. Conhecimento empresarial incorporado;
- 16. Mal compreendido pelos mantenedores;

Ou seja, infelizmente a Hotmart está repleta de código legado e que nascem a cada dua mais. Para se ter uma ideia, veja a quantidade de repositórios que temos hoje no GitHub:

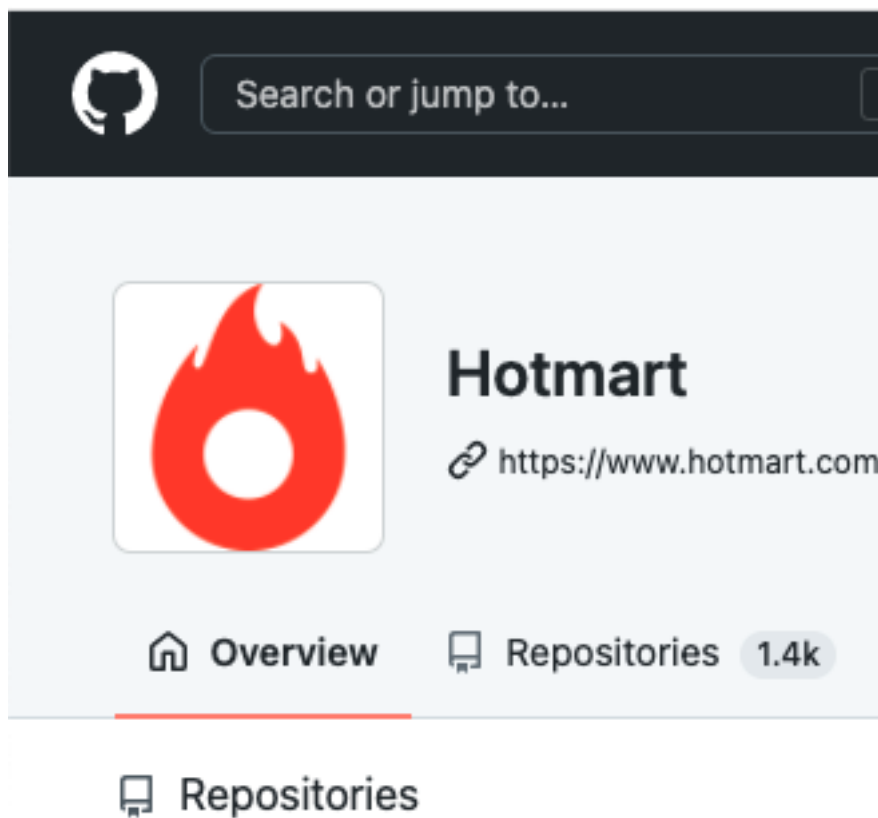


Figura 1: Repositórios presentes no GitHub

Obviamente nem todos esses repositórios representam uma APP ou API, mas se formos conservadores e considerarmos 50%, então teremos 700 repositórios onde nossas regras de negócio residem. Acho que levamos muito a sério a história de sair do monolito e acabamos criando tanto microserviço que as coisas podem ter saído um pouco do controle.

Mas então por que softwares legados existem? A resposta é simples: porque ninguém nasce sabendo aonde quer chegar. Uma empresa quando nasce, começa a ter que fazer suas escolhas. Ao longo do seu caminho, ela opta por boas ou não tão boas assim. Em um determinado momento, as escolhas passadas ajudam esta empresa a começar seguir em um caminho mais coeso. Isso se chama Maturidade.

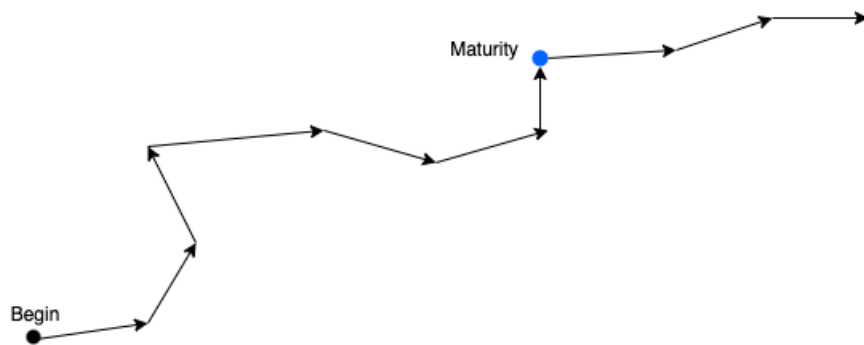


Figura 2: Processo de amadurecimento

Se fosse possível pegarmos tudo de bom e ruim que trilhamos e juntar as melhores escolhas desde o momento em que nascemos até o momento em que amadurecemos, seria perfeito. E é exatamente o que é possível fazer com um software: a consolidação do conhecimento.

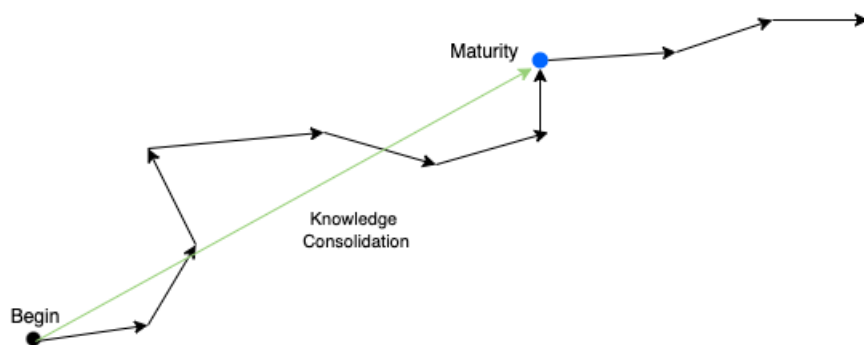


Figura 3: Processo de consolidação do conhecimento

Agora, mesmo que soubéssemos as melhores escolhas, de nada adiantaria chegar até aqui sem a correta manutenção de cada delas, ou seja, sendo que a cada nova decisão não fossem levadas em consideração todas as demais.

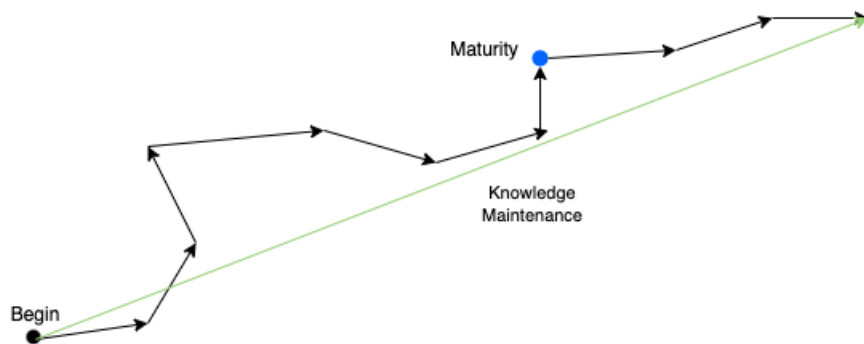


Figura 4: Processo de manutenção do conhecimento

Neste momento, estaríamos no que chamamos de “Estado da Arte” onde diversos setores da empresa poderiam trabalhar munidos de manuais para execução de seus trabalhos, deixando a parte criativa a ser utilizada no momento da criação de cada projeto.

Em outras palavras, durante todo o processo de reconstrução do software e dos processos internos, os próprios times iriam criando suas cartilhas ou manuais de trabalho de modo que quando fosse necessário algum incremento ou atualização, todos os envolvidos já saberiam como agir tanto na demanda quanto na avaliação por pares ².

Mas se isso parece ser tão interessante, por que não se faz?

2 Time em que está ganhando não se mexe

Acredito que uma das respostas mais óbvias é a de que “time em que está ganhando não se mexe”. Mas e se Messi, Neymar, Cristiano Ronaldo tivessem iniciado a carreira em um time que estivesse ganhando, o que seriam deles?. EU advogo que todo processo deve ser revisto tantas vezes quantas forem necessários pois:

1. você consegue melhorar o processo;
2. você melhora a percepção que você tem do processo;

Ou seja, sempre há uma melhora.

A análise e reconstrução leva tempo para se colher resultados, mas quando eles vêm, são muito significativos. Como uma amostra, há uma economia significativa de recursos computacionais e serviços de terceiros que poderíamos incorporar. Sem falar na redução drástica de tickets.

Não digo que a Hotmart nunca tenha feito mudanças nos códigos, mas clamo para que algum verdadeiramente profundo seja feito.

3 O Monte Everest

Outra frase muito ouvida é de que esse processo ou aquela informação é utilizada por tanta gente que me parece ser mais fácil Escalar o Monte Everest do que propor alguma mudança em meu contexto de trabalho.

4 O Poder do Conhecimento

Confesso que toda essa situação me deixava inquieto e após muita reflexão, tentei buscar na internet alguma resposta. Dentre o jargão corporativos norte-americano temos o padrão $C(x)O \mid x \in [A...Z]$ como por exemplo *CEO*³, *CTO*⁴, *CFO*⁵ e tantos outros.

Mas o que me chamou mais a atenção foi o **K**, da **CHIEF OF KNOWLEDGE OFFICER**. Dentre suas atribuições, temos⁶:

²Pull Request, por exemplo

³Chief of Executive Officer

⁴Chief of Technology Officer

⁵Chief of Financial Officer

⁶<https://searchcio.techtarget.com/definition/CKO>

Chief knowledge officer (CKO) is a corporate title for the person responsible for overseeing knowledge management within an organization. The CKO position is related to, but broader than, the CIO position. The CKO's job is to ensure that the company profits from the effective use of knowledge resources. Investments in knowledge may include employees, processes and intellectual property; a CKO can help an organization maximize the return on investment (ROI) on those investments. Furthermore, a CKO can help an organization to:

- Maximize the return on investment (ROI) in knowledge.
- Maximize benefits from intangible assets, such as branding and customer relationships.
- Repeat successes and analyze and learn from failures.
- Promote best practices.
- Foster innovation.
- Avoid the loss of knowledge that can result from loss of personnel.

Portanto, o que temos hoje como um mantra talvez pudesse formar o quarto pilar: O Pilar do Conhecimento. E afinal de contas, é muito mais sustentar qualquer coisa com 4 pilares do que com apenas 3.

Duas outras características bem interessantes do Knowledge Officer são:

- Manter todo o conhecimento centralizado e organizado
- Extrair o que as pessoas têm de melhor a oferecer

Ou seja, esse Officer deve estar sempre de braços abertos a toda e qualquer iniciativa, dando acolhimento e direcionamento adequado. e o melhor de tudo, poderemos reescrever a missão da empresa, veja como:

EXTRAIR O QUE AS PESSOAS TÊM DE MELHOR A OFERECER PARA PERMITIR QUE AS PESSOAS POSSAM VIVER DE SUAS PAIXÕES

Isso seria excelência de ponta a ponta.

Para finalizar, para que tudo isso possa ser possível, seria necessária a criação de uma nova Torre, chamada de Torre do Conhecimento onde agregaria todos os Especialistas, Product Managers, Designers e Writers. Essa Torre deveria receber as demandas da empresa, levantar requisitos, modelar entidades, elaborar layouts gráficos e textuais, e deve entregar um projeto no Jira já devidamente mensurado e organizado pronto para o time de desenvolvimento.

5 Metodologia para Reconstrução de Processos

Uma vez ouvi a seguinte frase dita por Gustavo Caetano⁷ que nunca mais saiu de minha mente:

⁷CEO da SambaTech

SE você me traz um problema
MAS não me traz a solução
ENTÃO você faz parte do problema

Depois de refletir por muito tempo, entendi que nem sempre sabemos a solução de todos os problemas assim quando nos deparamos com eles e que todos devem ser reportados imediatamente para mitigar quaisquer efeitos negativos. Por isso, adaptei essa frase para algo um pouco mais tangível:

SE você me traz um problema
ENTÃO vamos discutir uma solução
POIS juntos trabalhamos melhor

Baseado nessa frase, gostaria de apresentar uma metodologia ágil que estou desenvolvendo para reconstrução de processos maduros a qual batizei de **AIO-ROS**. Por mais caricata ou incomum que pareça, esta metodologia foi baseada na série japonesa de mangá e anime chamada **Os Cavaleiros do Zodíaco** veiculada no Brasil por volta dos anos 90.

Apesar de possuir diversas sagas, a primeira nos interessa por fazer uma analogia quase perfeita para o contexto que precisamos, onde os cavaleiros de bronze precisam salvar a princesa ferida, correndo contra o tempo, e para isso precisam passar pelas 12 casas do zodíaco enfrentando seus guardiões.

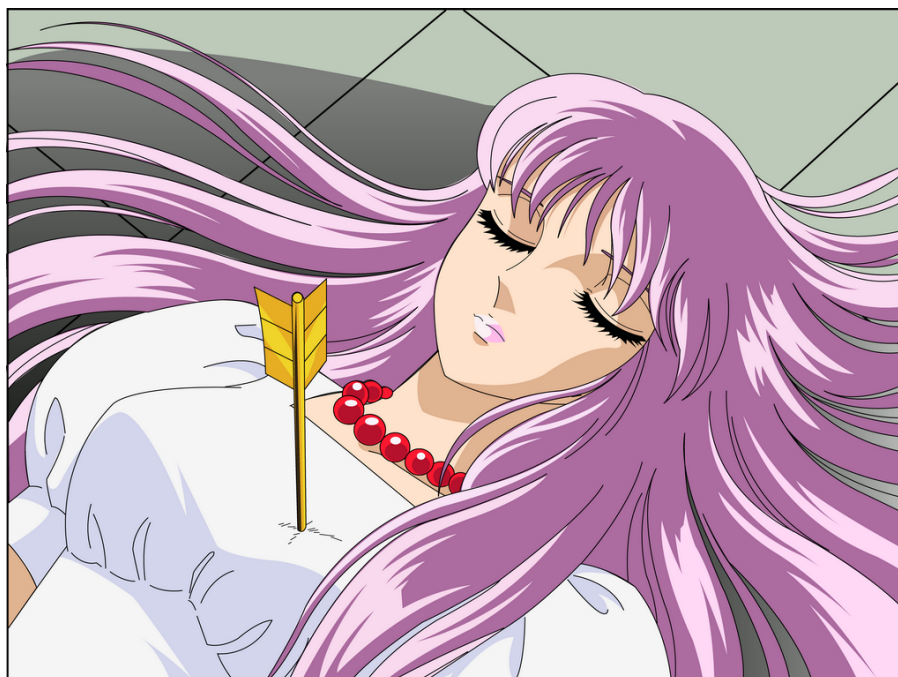


Figura 5: Saori, reencarnação de Athena, ferida

Adaptando essa história para nossa realidade, imagine que a princesa Saori seja a empresa ou o processo que está em execução de forma legada, ou seja, apenas com seu metabolismo basal não sendo capaz de entregar nada além do que suas funções básicas.



Figura 6: Os Cavaleiros de Bronze

Já os cavaleiros de prata são os funcionários que reconhecem essa fraqueza nos processos legados e lutam arduamente em suas casas, enfrentando um guardião por vez.

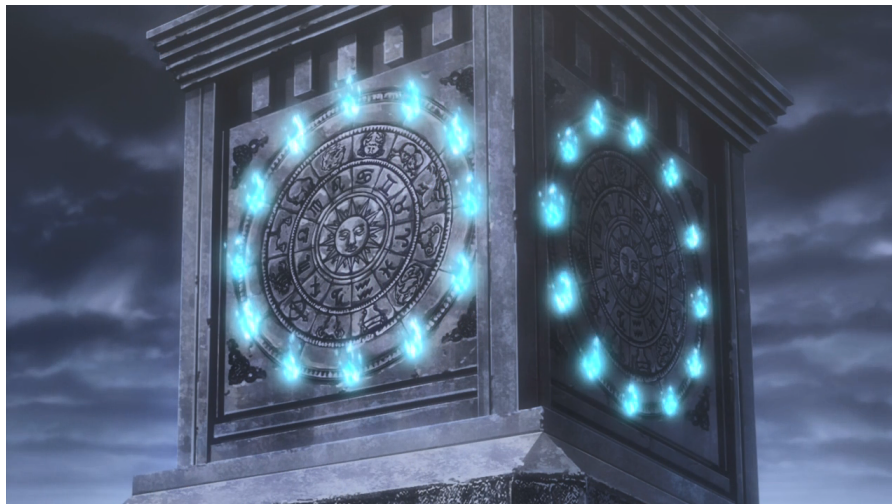


Figura 7: Relógio das 12 casas do Zodíaco

O Relógio do Zodíaco a agilidade, ou seja, o time não tem muito tempo a perder pois processos e clientes aguardam ansiosamente por melhorias e o time

deve trabalhar de forma integrada, coesa, a fim de garantir entregar dentro do prazo.



Figura 8: As 12 casas do Zodíaco

As 12 Casas do Zodíaco representam em quantas partes o processo como um todo deverá ser quebrado. Deste ponto de vista, faz mais sentido dizer **As n Casas do Zodíaco**. Mas aqui vale uma ressalva:

- Poucas casas deixam o sistema muito coeso, de difícil manutenção e com Cavaleiros de mais para combater
- Muitas casas deixar o sistema muito esparsa, de difícil manutenção e com Cavaleiros de menos para combater

O segredo é ter uma equipe experiente para se definir qual é o número ideal de casas.

Portanto, vamos ver como isso funcionaria na prática

1. Em uma reunião inicial a fim de explicar a todos os interessados o que será feito daqui pra frente;
2. Em uma segunda reunião, de preferência com os membros mais experientes, deve-se de definir quais serão as casas a serem atacadas bem como nomear seus guardiões⁸
3. Os guardiões, de posse de suas casas, devem definir os critérios de aceite e as regras de negócio que acharem pertinente à sua casa.
4. Agora é a vez dos cavaleiros utilizarem do que há de melhor, elevando seus cosmos ao sétimo sentido, propondo soluções engenhosas que capazes de cobrir tudo que foi proposto pelos guardiões daquela casa.
5. Neste momento, Product Managers, Designers e Writers atuam ativamente na composição da solução do produto.
6. Estes passos devem ser repetidos até que todo o projeto tenha finalizado.
7. Agora, Product Managers, Designers e Writers devem dar um formato no projeto, adicionando Epic, Estória, Tarefa, Sub-Tarefa para que os times executantes possam indicar o esforço para cada tarefa descrita.
8. Writers, não se esqueçam de que faz parte da entrega todas as chaves estarem traduzidas quando o projeto estiver fechado, ou seja, o time de desenvolvimento não poderá iniciar qualquer tarefa desde que essa condição não seja satisfeita. Essa entrega pode ser feita até via arquivo de texto anexado no card do jira, por exemplo.
9. Voltando a falar sobre mensurar, como já teremos nosso livro de referência, teremos uma grande noção do quanto dura cada atividade, pelo menos uma boa aproximação. Dessa forma, fica bem mais simples e preciso saber quando uma tarefa deverá ser finalizada.

⁸deve-se haver pelo menos 2 guardiões para que haja uma cooperação mútua