



(<http://luizricardo.org/>)

Arquitetura (<http://luizricardo.org/categoria/arquitetura/>) / Desenvolvimento (<http://luizricardo.org/categoria/desenvolvimento/>)

/ Carreira (<http://luizricardo.org/categoria/carreira/>) / Outros (<http://luizricardo.org/categoria/assuntos-randomicos/>)

/ Sobre (<http://luizricardo.org/sobre/>)

Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 1 – The Tar Pit (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/>)

📅 14/12/2015 (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/>) ➔ Assuntos Randômicos (<http://luizricardo.org/categoria/assuntos-randomicos/>)

Em 1975, Frederick P. Brooks escreveu *The Mythical Man-Month* (O Mítico Homem-Mês), um dos livros mais conhecidos, revolucionários e controversos sobre desenvolvimento de software.

Em seu livro, Brooks reúne grande parte de seu conhecimento sobre Engenharia de Software. Baseado em conceitos fundamentais da computação e na experiência prática de ter gerenciado um dos maiores projetos de desenvolvimento de sua época, o autor fez várias afirmações sobre o então presente e futuro do desenvolvimento de software.

Vinte anos depois, em 1995, seu livro ganhou uma nova edição e capítulos adicionais. Um deles revisa de forma sistemática cada proposição do autor e acrescenta comentários sobre sua validade. A questão é: após anos de prática de Engenharia de Software, permanecem as afirmações consistentes ou foram desmistificadas por novos conhecimentos, descobertas e experimentos? Em outro capítulo, o autor novamente revisa os argumentos principais do livro. Ele reconhece quando estava errado? Quais argumentos estão obsoletos?

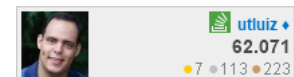


FOLLOW

Novos artigos por e-mail. **Sem spam!**



Acompanhe minhas respostas no Stack Overflow em Português.



(<http://pt.stackoverflow.com/users/227/utl Luiz>)

POPULAR

- Instalando, configurando e usando o Maven para gerenciar suas dependências e seus projetos Java (<http://luizricardo.org/2014/06/instalando-configurando-e-usando-o-maven-para-gerenciar-suas-dependencias-e-seus-projetos-java/>)
- Um guia prático para passar em entrevistas para desenvolvedor de software (<http://luizricardo.org/2015/03/um-guia-pratico-para-passar-em-entrevistas-para-desenvolvedor-de-software/>)
- Construindo objetos de forma inteligente: Builder Pattern e Fluent Interfaces (<http://luizricardo.org/2013/08/construindo-objetos-de-forma-inteligente-builder-pattern-e-fluent-interfaces/>)

Hoje, **quarenta anos depois da primeira edição de *The Mythical Man-Month***, gostaria de novamente revisar as afirmações e os argumentos de Brooks. Juntos, vamos então compará-las com as práticas modernas da Engenharia de Software.

Como vai funcionar

Irei publicar um artigo para cada capítulo do livro, começando a partir deste que inclui o capítulo 1.

Irei descrever os principais argumentos do autor em cada um dos capítulos. Usarei como guia o capítulo 18 da edição de aniversário de Mythical Man-Month, *Propositions of The Mythical Man-Month: True or False?*, que já contém um resumo das proposições feitas pelo autor.

Para cada proposição principal, adicionarei comentários ou explicações, de acordo com a obviedade ou não da mesma. O leitor notará que muitas das proposições são obviamente verdadeiras ou auto-explicativas. Não adentrarei em detalhes nesses casos.

Ao final de cada artigo, farei uma síntese sobre os pontos, destacando os que ainda são válidos hoje, parcialmente válidos (dentro de certo contexto, por exemplo), os que já foram ultrapassados pelo tempo ou que não se mostraram verdadeiros.

Capítulo 1: *The Tart Pit*

Neste capítulo, Brooks inicia sua argumentação com uma analogia: o desenvolvimento de grandes sistemas de software é tão difícil como foi para os grandes e pesados animais da pré-história libertarem-se de um poço de piche, muitos sucumbiram e foram tragados. Quantos projetos grandes e pequenos, com equipes massivas ou enxutas, não foram abandonados ou cancelados sem conseguir avançar para seu objetivo?

” *É necessário 3 vezes mais esforço para desenvolver um componente de software em relação a um software para uso privado, assim como 3 vezes mais esforço para transformá-lo em um produto coerente e testado.*

É muito difícil, se não impossível, avaliar a afirmação acima com exatidão. O bom senso diz que isso varia de acordo com as características do projeto. Entretanto, é patente que o princípio por detrás dela é pura verdade.

Por exemplo, um desenvolvedor pode criar um script para uso pessoal muito rapidamente. Mas se ele quiser publicar aquele script para os colegas reusarem, vai precisar modificá-lo para atender casos mais genéricos e parametrizar as informações que era fixas em um ambiente. Talvez ele precise definir uma API, escrever alguma documentação ou mesmo uma GUI consistente. Outro passo seria transformar isto numa ferramenta para uso de qualquer interessado em outras empresas e ambientes. Um produto de software vai demandar muito mais esforço.

Particularmente eu diria que o fator de 9 vezes para transformar um software privado em um produto coerente é ainda conservador, se considerarmos todos os aspectos como hospedagem e suporte. Testes automatizados ou manuais também somam-se ao *custo da qualidade* do produto.

” *A profissão de programar envolve gratificação pessoal: a alegria de construir as coisas.*

- O que é um sistema legado? (<http://luizricardo.org/2013/09/o-que-e-um-sistema-legado/>)
- Instalando e Configurando o Java Development Kit 7 para Desenvolvimento (<http://luizricardo.org/2013/11/instalando-e-configurando-o-java-development-kit-7-para-desenvolvimento/>)
- Morando no interior, trabalhando na capital: experiências das viagens de fretado (<http://luizricardo.org/2013/12/morando-no-interior-trabalhando-na-capital-experiencias-das-viagens-de-fretado/>)
- Entendendo o Triângulo de Ferro, porque não podemos ter tudo (<http://luizricardo.org/2013/09/entendendo-o-triangulo-de-ferro-porque-nao-podemos-ter-tudo/>)

DESCUBRA

Selecionar categoria

Selecionar o mês

- ◀ agile (<http://luizricardo.org/tag/agile/>)
- ◀ ambiente de trabalho (<http://luizricardo.org/tag/ambiente-de-trabalho/>)
- ◀ apostila (<http://luizricardo.org/tag/apostila/>)
- ◀ apresentação (<http://luizricardo.org/tag/apresentacao/>)
- ◀ automação (<http://luizricardo.org/tag/automacao/>)
- ◀ boas práticas (<http://luizricardo.org/tag/boas-praticas/>)
- ◀ bootstrap (<http://luizricardo.org/tag/bootstrap/>)
- ◀ css (<http://luizricardo.org/tag/css/>)
- ◀ curso (<http://luizricardo.org/tag/curso/>)
- ◀ desempenho (<http://luizricardo.org/tag/desempenho/>)
- ◀ desenvolvimento profissional (<http://luizricardo.org/tag/desenvolvimento-profissional/>)
- ◀ design patterns (<http://luizricardo.org/tag/design-patterns/>)
- ◀ dica (<http://luizricardo.org/tag/dica/>)
- ◀ eclipse (<http://luizricardo.org/tag/eclipse/>)
- ◀ emprego (<http://luizricardo.org/tag/emprego/>)
- ◀ engenharia de software (<http://luizricardo.org/tag/engenharia-de-software/>)
- ◀ estimação (<http://luizricardo.org/tag/estimacao/>)
- ◀ estudo (<http://luizricardo.org/tag/estudo/>)
- ◀ estudo de caso (<http://luizricardo.org/tag/estudo-de-caso/>)
- ◀ framework web (<http://luizricardo.org/tag/framework-web/>)
- ◀ gerenciamento de projetos (<http://luizricardo.org/tag/gerenciamento-de-projetos/>)
- ◀ html5 (<http://luizricardo.org/tag/html5/>)

A afirmação nunca foi tão verdadeira.

Vemos movimentos como *agile* e *software craftsmanship* reforçando esta ideia. A experiência cotidiana também.

Os melhores desenvolvedores são pessoas que amam o que fazem. Embora até possam existir gênios que aprendem sem muito esforço, aqueles que mais se destacam são os que acreditam no que fazem e se esforçam diariamente.

” *A profissão também envolve sofrimento*

Brooks afirma que programadores precisam se adaptar ao nível de perfeição exigido, isto é, programar software sem erros pois o computador não tolera pequenas falhas, o que é muito difícil e penoso no começo.

Programadores em geral têm seus objetivos definidos por outros e depende de outras pessoas para atingi-lo. Eles não podem controlar a situação e não tem autoridade necessária para fazer as coisas acontecerem na grande maioria dos casos.

Brooks ainda lembra que para cada momento de criatividade para projetar algo é posteriormente acompanhando por horas e mais horas de trabalho desgastante para atingir o objetivo. É como quando assistimos a uma conferência e saímos cheios de ideias, mas o trabalho para colocar tudo em prática nos desestimula completamente.

O projeto de software converge mais de vagar na medida em que chega ao fim. Muitos pensam que é possível acelerar o desenvolvimento com a experiência que a equipe acumula no decorrer do projeto, mas em geral ocorre o oposto, já que muitas atividades ocultas são descobertas, bugs se acumulam e a base de código e a complexidade crescem.

Por fim, muitos dos produtos de software são ameaçados de obsolescência antes mesmo de estarem completos. O motivo é que as necessidades mudam constantemente e qualquer implementação começa a se tornar obsoleta no momento em que foi realizada. Ajustamos constantemente o nosso entendimento e até o projeto de um sistema, sendo que geralmente não é possível evoluir as implementações na mesma velocidade.

Considerações sobre o capítulo

Este é apenas um capítulo introdutório sobre pontos gerais relacionados aos problemas do desenvolvimento de software e ao ofício relativo a ele. Talvez por isso, em maior ou menor grau, todos os pontos são tão ou mais relevantes e verdadeiros hoje em relação a 1975.

Essa conclusão com certeza dá um gosto especial para apreciarmos os demais capítulos, para os quais publicarei comentários em breve.

Não percam! 😊

← engenharia de software (<http://luizricardo.org/tag/engenharia-de-software/>)

← mythical man-month (<http://luizricardo.org/tag/mythical-man-month/>)



Luiz Ricardo (<http://luizricardo.org/author/utluizgmail-com/>)

Ex-engenheiro, arquiteto, agilista, artesão, agora dedica-se a desenvolver software bem.

Share!

← javascript (<http://luizricardo.org/tag/javascript/>)

← jquery (<http://luizricardo.org/tag/jquery/>)

← maturidade (<http://luizricardo.org/tag/maturidade/>)

← maven (<http://luizricardo.org/tag/maven/>)

← meme (<http://luizricardo.org/tag/meme/>)

← monografia (<http://luizricardo.org/tag/monografia/>)

← mythical man-month (<http://luizricardo.org/tag/mythical-man-month/>)

← oracle (<http://luizricardo.org/tag/oracle/>)

← play! framework (<http://luizricardo.org/tag/play-framework/>)

← programação (<http://luizricardo.org/tag/programacao/>)

← python (<http://luizricardo.org/tag/python/>)

← reflexões (<http://luizricardo.org/tag/reflexoes/>)

← requisitos (<http://luizricardo.org/tag/requisitos/>)

← segurança (<http://luizricardo.org/tag/seguranca/>)

← sql server (<http://luizricardo.org/tag/sql-server/>)

← stackoverflow (<http://luizricardo.org/tag/stackoverflow/>)

← sysdeo (<http://luizricardo.org/tag/sysdeo/>)

← tdd (<http://luizricardo.org/tag/tdd/>)

← testes (<http://luizricardo.org/tag/testes/>)

← tomcat (<http://luizricardo.org/tag/tomcat/>)

← transação (<http://luizricardo.org/tag/transacao/>)

← tutorial (<http://luizricardo.org/tag/tutorial/>)

← ubuntu linux (<http://luizricardo.org/tag/ubuntu-linux/>)


SOCIAL





(<http://github.com/utluiz/>)





(<http://linkedin.com/in/utluiz/>)

 Facebook 7 (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/?share=facebook&nb=1>)

 Reddit (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/?share=reddit&nb=1>)

 Twitter (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/?share=twitter&nb=1>)

 LinkedIn (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/?share=linkedin&nb=1>)

 Google (<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-1-the-tar-pit/?share=google-plus-1&nb=1>)

 Mais

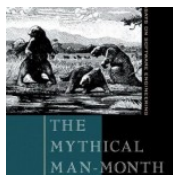
[ANTERIOR](#)

[PRÓXIMA](#)

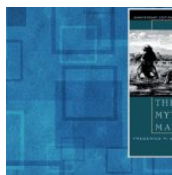
Porque não acompanho mais o Stack Overflow em Português
(<http://luizricardo.org/2015/12/porque-nao-acompanho-mais-o-stack-overflow-em-portugues/>)

Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 2 – The Mythical Man-Month
(<http://luizricardo.org/2015/12/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-2-the-mythical-man-month/>)

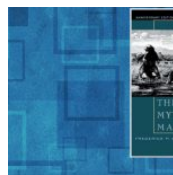
Related



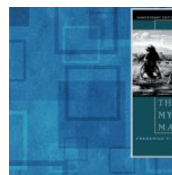
(<http://luizricardo.org/2017/05/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-6-passing-the-word/>) Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 6 – Passing the Word



(<http://luizricardo.org/2017/05/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-3-the-surgical-team/>) Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 3 – The Surgical Team



(<http://luizricardo.org/2017/01/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-7-why-did-the-tower-of-babel-fail/>) Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 7 – Why Did the Tower of Babel Fail?



(<http://luizricardo.org/2017/07/revisitando-as-proposicoes-de-brooks-em-the-mythical-man-month-capitulo-10-the-documentary-hypothesis/>) Revisitando as Proposições de Brooks em The Mythical Man-Month – Capítulo 10 – The Documentary Hypothesis