

Parte 1 - Software Engineering Laws for Information Systems in Everything (Big and Small Systems)

FU (Fixed and Updated)

Todo sistema, seja grande ou pequeno, deve ter provisões que permitam que esse sistema seja consertado ou passe por atualizações, seja de dentro do sistema ou fora dele.

CK (Contextual Knowledge)

Todo sistema, seja grande ou pequeno, deve ter provisões, que permitam que o sistema adquira e faça uso de conhecimento contextual pra decidir se ele está funcionando apropriadamente ou não. (exemplo: a fechadura digital da sua casa deve ter conhecimento contextual de uma década operando)

FUCK property

Todo sistema que obedece as leis FU e CK pode verificar as condições do seu ambiente operacional pra decidir se suas ações são coerentes ou não com o contexto daquele ambiente, Mas ainda, se ele decidir que elas não são, ele pode se auto-consertar (voltar pra um estado coerente) ele pode pedir por external help e também pode ser consertado por algum agente externo.

IN (independece of network)

Todo sistema, grande ou pequeno, deve conter previsões para para se manter independente de rede o tanto quanto possível (exemplo: um aplicativo de notícias, ao não possuir conexão com a internet, mostra notícias do último momento com internet). Na ausência de rede, deve haver uma degradação para um nível de utilidade que poderá lembrar um sistema totalmente independente e isolado.

G (general)/ SIFU property

Em geral, para todo tipo de sistema, segurança e integridade devem ser tratados com importância maior do que funcionalidade, que por sua vez deve ser balanceada com a usabilidade.

FUCKING property

Se um sistema obedece as leis FU, CK, IN, e G e está quase sempre em rede (e possui uma performance aceitável sem network) ele pode verificar as condições do seu ambiente operacional e decidir se suas ações são coerentes com o contexto daquele ambiente. Ainda por cima, se ele decidir que as ações não são coerentes, ele pode se consertar, pedir para ser consertado, ou esperar pelo conserto de algum agente externo. Em todos os casos, segurança, integridade, funcionalidade e usabilidade devem estar apropriadamente balanceadas.

REAL laws

R - Todo sistema deve ter provisões que permitirão seu re-uso em diferentes contextos e diferentes sistemas.

E - Todo sistema deve ter provisões para ser estendido a para acomodar novos comportamentos.

A - Todo sistema deve ter provisões para oferecer análise de dados, para que seja medido o que está acontecendo e como ele está evoluindo eventualmente.

L - Todo sistema deve possuir provisões que o permitam ser fracamente acoplado, para reduzir a dependencia com outros sistemas e também a interdependencia com seus proprios módulos e componentes.

Parte 2 – Traçando paralelos com o projeto PromoKING

Conhecendo todas as leis e suas propriedades, podemos traçar paralelos com o projeto referência para a disciplina. No decorrer do desenvolvimento do software, é muito fácil acabar esquecendo de algum ponto específico o qual as leis abrangem. Nesse aspecto, esse trabalho visa apontar caminhos o qual podemos tomar durante o desenvolvimento para atender todas as leis e ainda, os possíveis riscos de não segui-las adequadamente.

FU law – O próprio processo de documentação contribui para uma boa manutenção do software. As atualizações futuras estão documentadas na forma de requisitos “desejáveis”. A manutenção e suporte do software será

contínua, além disso, como seus módulos e pacotes estarão bem divididos e independentes, sua manutenção será facilitada.

CK law – O nosso aplicativo, que é baseado num feed de recomendações, contará com uma opção de “não tenho interesse” para que então haja melhora constante do aplicativo e a performance do algoritmo. Também contaremos com uma seção de “fale conosco” para atender a quaisquer dúvidas ou problemas que possam surgir. Também surgirão questionários opcionais e rápidos para que o cliente demonstre como está sendo sua experiência com o aplicativo.

Resumidamente, nosso aplicativo atenderá a FUCK property pois o algoritmo de recomendação tratará de melhorar seu desempenho continuamente com base da opinião e interação do usuário com os produtos.

IN law – Naturalmente, o aplicativo não funcionará corretamente sem o uso de internet. Contudo, serão mostrados anúncios e publicações salvos durante a última sessão. No topo do feed, haverá um aviso informando ao usuário que ele está sem conexão, e que os anúncios podem não estar atualizados.

G law – Questões de segurança e integridade de dados ainda estão sendo debatidos entre a equipe de desenvolvimento, devido o problema da falta de conhecimento sobre esses aspectos. Após realizado nosso levantamento bibliográfico sobre o tema, trataremos de incluir as melhores soluções possíveis referente a segurança e integridade dos dados dos nossos clientes.

FUCKING property – Resumidamente, o algoritmo cuidará para que seja oferecida a melhor experiência possível para os clientes, tomando decisões com base em suas atividades. Os questionários de satisfação cuidarão para que o melhoramento do algoritmo também ocorra por influências externas.

R law – A disposição dos pacotes garantirá uma certa independência entre eles e suas funcionalidades, permitindo uma rápida adaptação para um possível re-uso.

E law – Apesar de ainda estarmos levantando informações sobre a maneira como atualizações são aplicadas num sistema, contamos com o foco na

independencia e no loosely coupled para que ao implementar quaisquer tipos de atualizações, não tenhamos muitos problemas para adição de novas funcionalidades.

A law – Todos os dados gerados serão armazenados. Buscaremos fazer análises sob diferentes aspectos, seja para melhora do aplicativo como a também a melhora de como os estabelecimentos comerciais podem ofertar seus produtos aos seus clientes.

L law – O processo de desenvolvimento do aplicativo está sendo realizado com base de pacotes 4 layers (domínio, GUI, negocio, persistência). Essa divisão, juntamente com a separação de pacotes dos componentes, permitirá um fraco acoplamento garantido então uma baixa interdependência entre esses.