

# FSMで うれしいこと

あ、長久です  
念のため

# あらためて FSMで うれしいこと

# 使用前

# その1

# 仕様書は 箇条書き

キャラの挙動  
が膨大な箇条  
書きの集合で  
表現されてる

箇条書きの  
項目間に不整  
合があっても  
分からない

# 実装・試験 フェーズで 露呈する

# あいつの 仕様書 使えねえな

# その2

# 仕様書はフ ローチャート でそれっぽく

いわゆるフ  
ローチャート  
は表現力が足  
りない

表現し切れ  
ない部分に  
不具合があ  
っても分  
からない

# 実装・試験 フェーズで 露呈する

# あいつの 仕様書 使えねえな

# 使用後

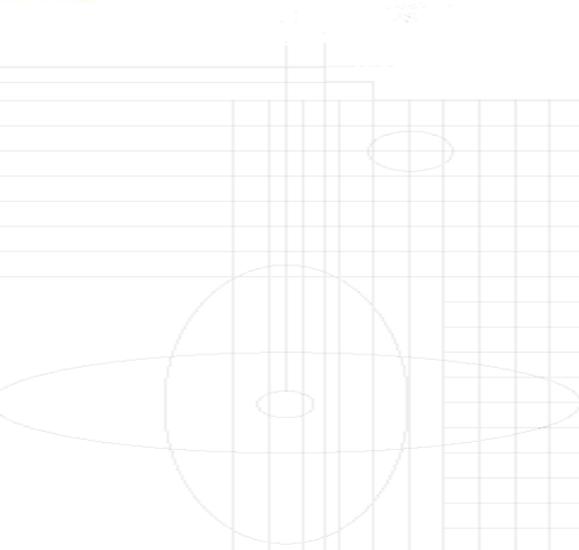
# 仕様書は UML

キャラの挙動  
が全て連結さ  
れている

十分な表現能力で細かいところまで書ける

悪いところが  
あると図に  
ならない

# そのまま実装 できる



# GJ



# なんでだろう

# 企画者 日本語で 考える

# 技術者 コンピュータ の立場で 考える

# 越えられない 溝

でも

企画者は日本語  
でしか考え  
られないわけ  
じゃない

# 論理思考のサ ポートと見え る化をUMLで 行う

# 実装可能な仕様 が表現でき る

# 実装前にし ビュアーでできる

# 技術者が いなくてももし ビュアーでできる



技術者は  
家に帰ったり  
寝たりできる

技術者は  
重要な作業に  
時間を使える

企画者も技術者もうれしい

# もっとうれし い未来の話

# モデル検査にか けられる

# 定理証明にか けられる

# おまけ

# RBS と FSM



# RBS: ぶっちやけ 状態遷移ルー ルの集合

# 展開しない分 FSMよりラク ちゃんけ

が、しかし

いつ  
どのルールが  
発火するのか  
分かり難い

# FSMの方が レビューや デバッグが しやすい

# FSMとRBSは 等価

実装や開発工  
程に合わせて  
選べば良い

おしまい