

Q & A: Groovy at Mutual of Omaha



Groovy After All These Years
Or
Why Your Company Should Adopt Groovy!

Groovy in 2005

- groovy-1.0-beta-9 (2 years before 1.0 release)
- No plugin support in any of the main IDEs
- The Groovy Project was gaining momentum
- The Groovy Community was very welcoming and supportive

Why Mutual Adopted Groovy

- Ran in JVM
- Worked seamlessly with larger Java Project
- Handled Big Decimal Calculations
- Very Readable Code
- Easy for non-Java programmers to adopt

The Development Team

- Developed by a team of 8
 - 1 Project Manager
 - 3 Members of the Group Actuarial Staff
 - 4 I/S Developers
- Extremely Limited Java Experience
 - 1 Seasoned Java Developer
 - 1 developer with some Java experience
 - 5 with No Actual Java development experience

Code Readability (Process)

```
package com.mutualofomaha.product
```

```
class ProductRateCalc {  
  def createProductOutput(productInput) {  
    def productOutput = new ProductOutput()  
    def planFactors = [ ]  
    productInput.planDesigns.each{pd ->  
      def planCensus = productInput.planDesignCensusMap.get(pd.planDesignId)  
      def members = generatePlanMembers(planCensus)  
      calculateMemberFactors(members)  
      • def planFactor = generatePlanFactors(members)  
      • planFactors << planFactor  
      calculateExpenses(planFactor, members)  
    }  
    planFactors << generateAlternateFactor(planFactors)  
  }  
}
```

Code Readability (Calcs)

```
billingClaimCost = premium - (  
    typeAmt +  
    typePerEe +  
    typePctCc +  
    comsData.comsAmt +  
    ( comsData.comsPepm * planDesigns[0].altPartLivesAmt * 12 ) +  
    type2Flat +  
    miscExp)  
  
members.each{member ->  
    member.eeDisClm = member.eeVol * member.factor1 *  
        planFactor.factorA * planFactor.factorB * planFactor.factorC *  
        member.factor2 * member.factor3  
}
```

Deployed to Production in 2008

- Working Rating Engine
- A Versatile Test Harness
- 45,000 lines of Code
- Avg time to rate/price was 3 to 4 seconds (not all of which was rating engine time)

Since 2008

- Other Developers have come and gone
 - 5 additional Group Actuarial individuals have been or are active in the code
 - 7 additional I/S Developers have been or are active in the code base
 - Only 2 of these 12 had significant Java experience prior to supporting the Rating Engine

Since 2008

- The Code base has Grown Significantly
 - Added 3 New Products (and more on the way)
 - Re-architected the calculations for almost every product
 - Now at 120,000 Lines of Code
 - Developed an Entire New Reporting Process in Groovy
 - Ported from DB2 to Teradata

Run Time Errors

- Run Time Errors are a Concern
 - We've had less than 10 instances in prod
 - They can be costly and embarrassing
- BUT Not as big a concern as failing silently!
 - Calculation errors
 - Calculation omissions
 - Requirements misunderstandings

Special Thanks to ...

- Guillaume Laforge
- John Wilson
- Jochen Theodorou
- The Rest of the Groovy Community!