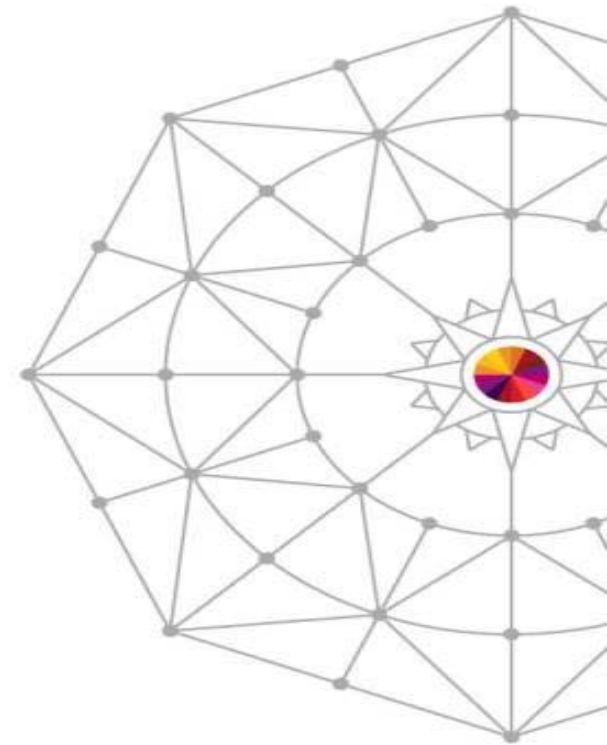




# Business Agility – Unlocking Your Legacy Code

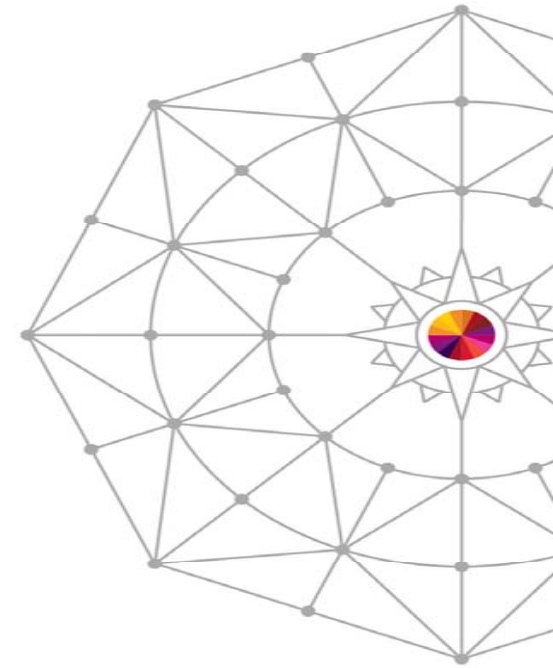
Speaker Name **John Rhodes & Denise Kalm**  
Speaker Company **CM First Group**

Date of Presentation **Thu March 13, 2014**  
Session Number 14788

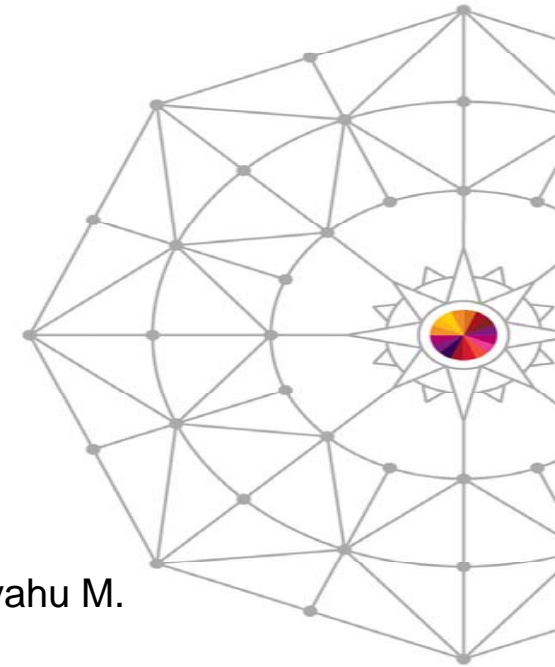


# Business Agility

- **Agility** is a concept that incorporates the ideas of flexibility, balance, adaptability, and coordination under one umbrella.
- In a business context, **agility** typically refers to the ability of an organization to rapidly adapt to market and environmental changes in productive and cost-effective ways.



# Is software development / technical debt your organization's constraint?



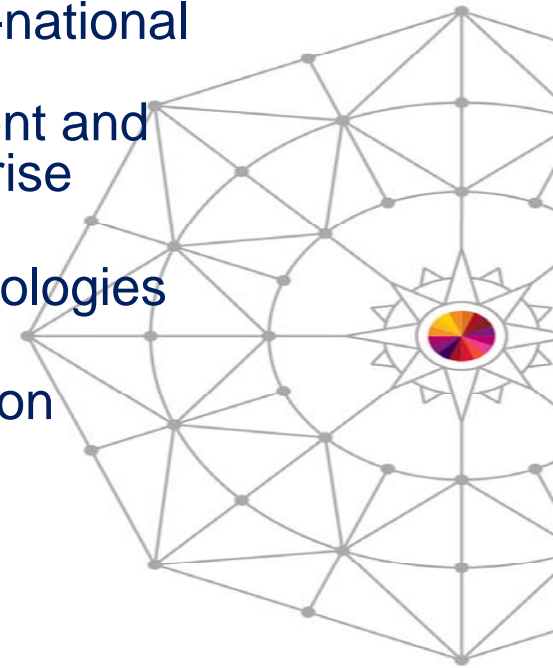
Theory of constraints (TOC) is an overall management philosophy introduced by Eliyahu M. Goldratt.

**"A chain is no stronger than its weakest link."**

## Speaker Bio



- CTO of CM First Group, a multi-national software and services company focused on software development and modernization of IBM i/z enterprise solutions
- Prior experience with CA Technologies and Kraft Foods
- Speaker at IBM and CA events on Legacy Modernization
- From Austin, Texas



## Speaker Bio



- Chief Innovator of Kalm Kreative, Inc.
- Consultant to CM First
- Prior technical and pre-sales experience with CA, BMC Software, Cybermation and various customer sites
- Speaker at SHARE and CMG
- From Walnut Creek, CA



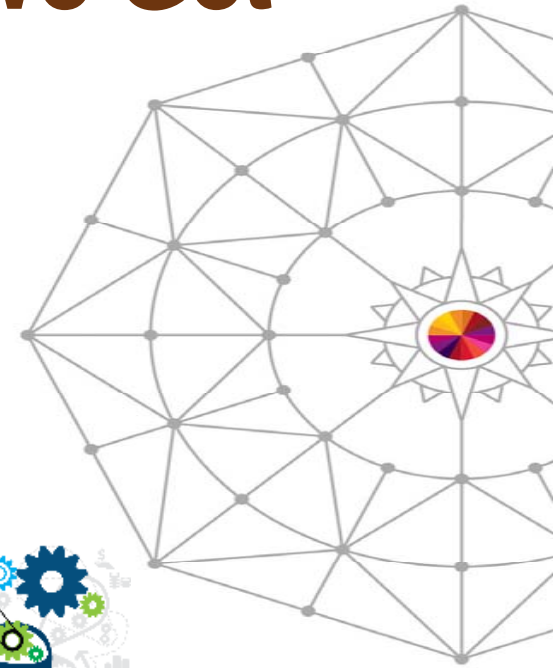


# Agenda

- Business Challenges
  - How was Agility lost?
  - Costs of Clumsiness
- Legacy Code Issues
  - What's There?
  - Dead End Frameworks
  - Technical Debt
- Solutions
  - Dynamic, Continuous Assessment using Enterprise Metadata
  - Remediate Tech Debt
    - Clones
    - Maintainability
    - ...
  - Get in a position to Modernize

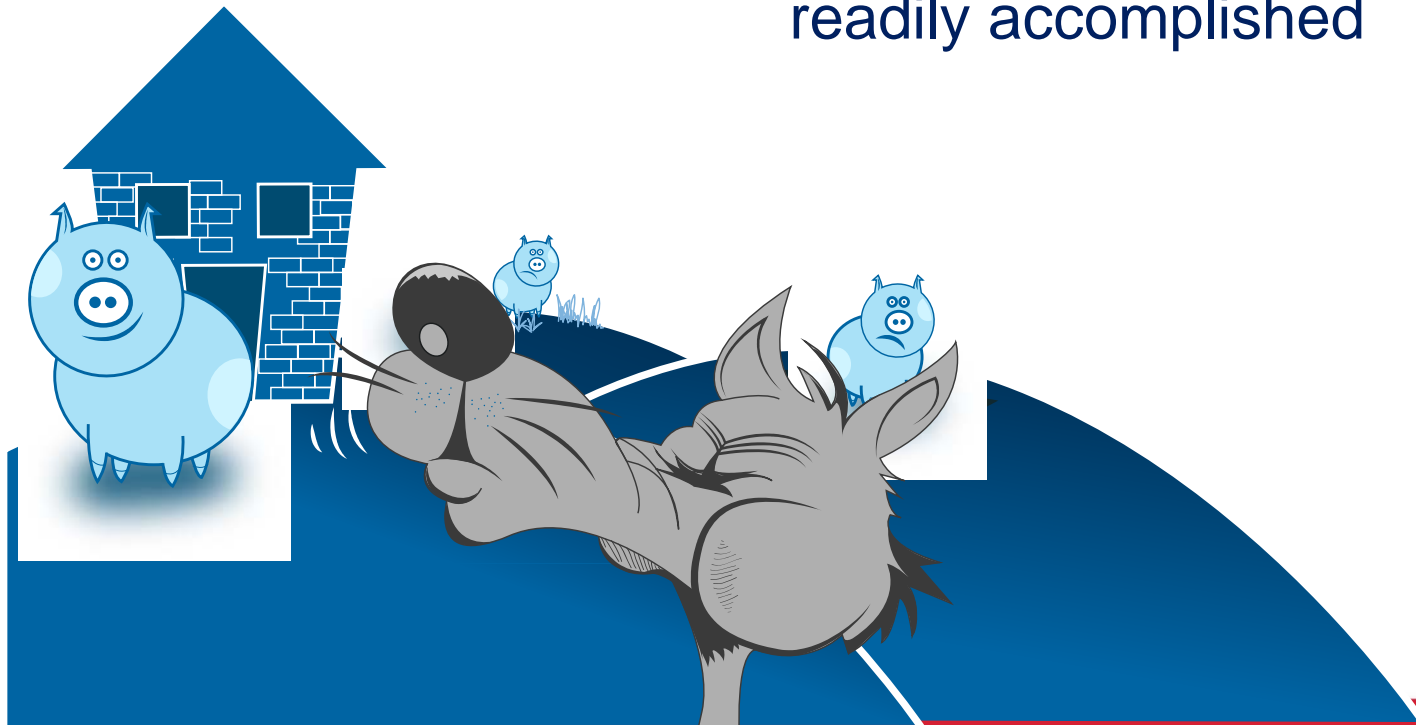


# How Did We Get Here?



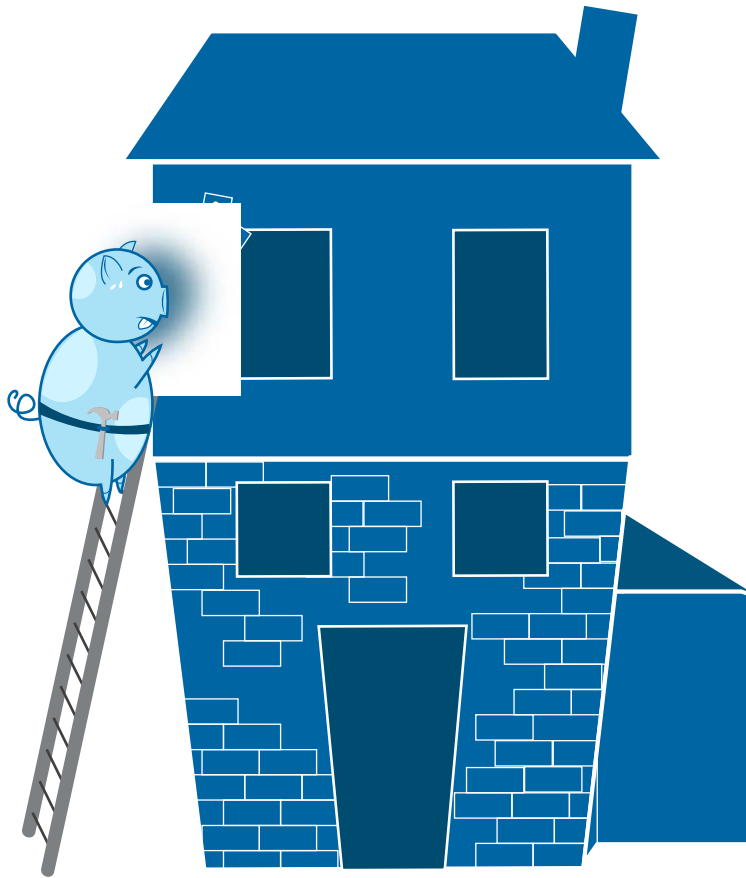
## Back in the day, agility came easily

- Adherence to Design
- Cohesive Technical Team
- Enhancements and Fixes were readily accomplished

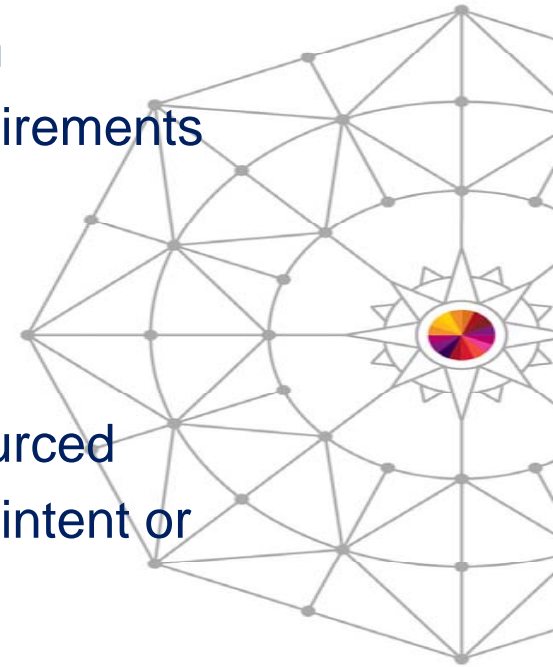




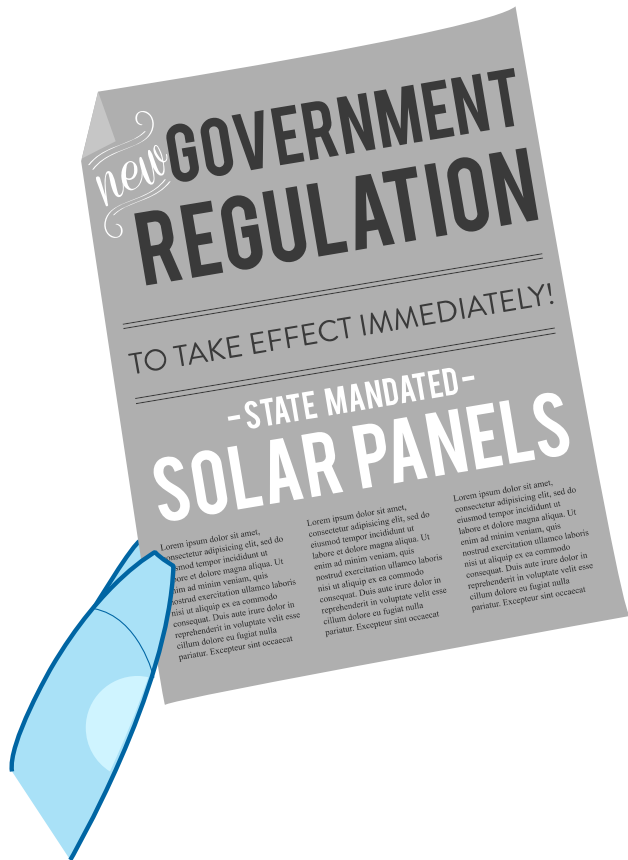
# Technical Debt Grows over Time, Reducing Agility



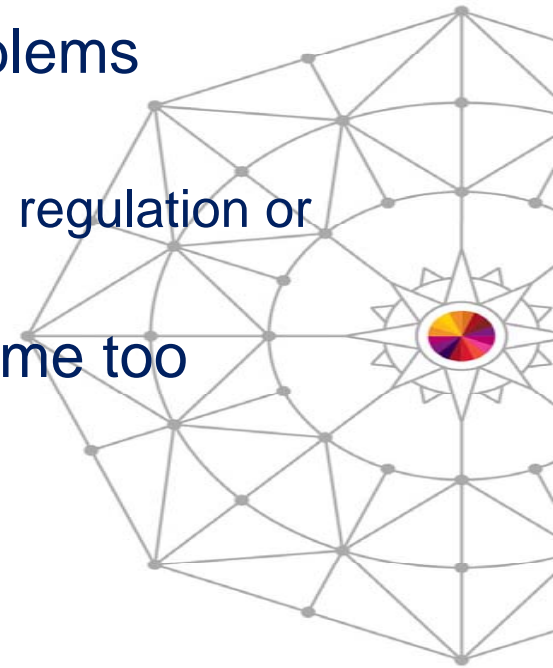
- Short Cuts Taken
  - Layering on Requirements
  - Tight Deadlines
  - Lack of Architects
- Different Teams
  - Potentially Outsourced
  - Don't understand intent or architecture



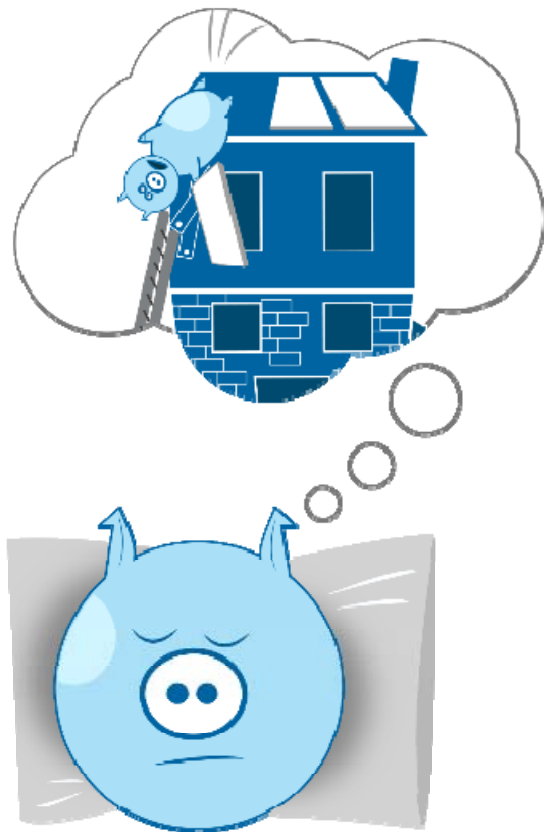
# Until you reach a point where...



- Major Change Problems
  - New Technology
  - Adapting to external regulation or factors
- The code has become too brittle...

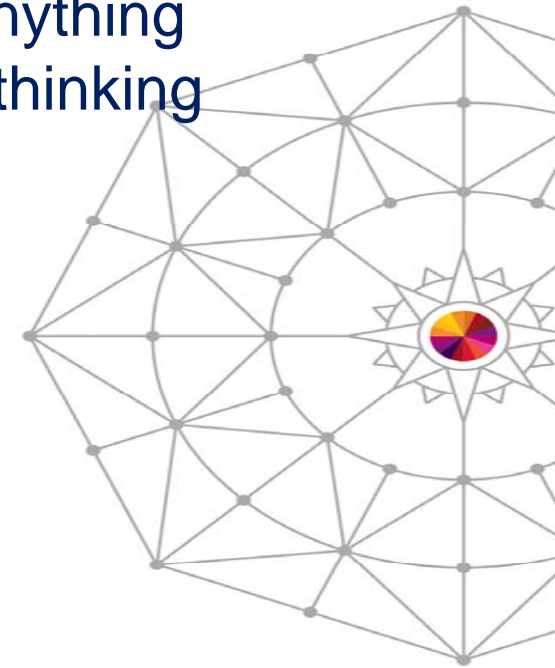


# Disaster is at hand



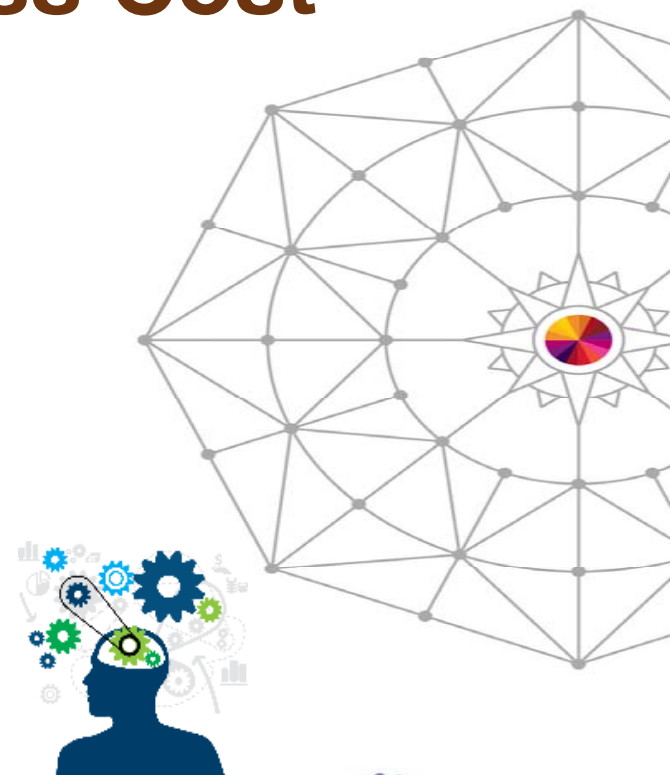
Everyone is afraid to do anything  
– and can't get any sleep thinking  
about it!

Which is the opposite  
of Agility...





# Business Cost



Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)



# Technical Debt / Quality Cost



**Fig. 1 - The Cost of Technical Debt**

- Increased
  - Analysis Costs
  - Time to Market
  - Risk
- Decreased Customer Satisfaction



# Analysis Costs



- Outdated documentation and lack of knowledge pervasive in all systems
- Analysis Cost High
  - **45-60% of time spend finding/quantifying what to do**
- Little time to spend on new business
- Outages, performance problems



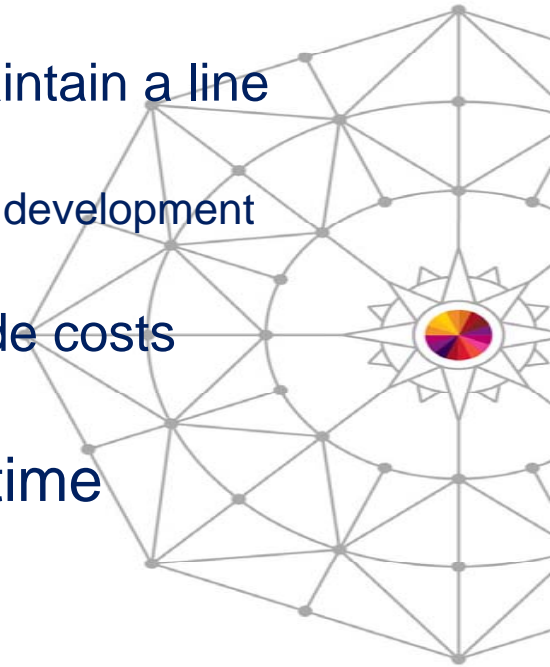
Facts and Fallacies of Software Engineering - Glass



# Coding Costs



- Coding Cost
  - Cost \$1 / year to maintain a line of code
    - Start to dwarf original development costs
  - And poor quality code costs much, much more
- Coders can spend time developing



# Opportunity Costs



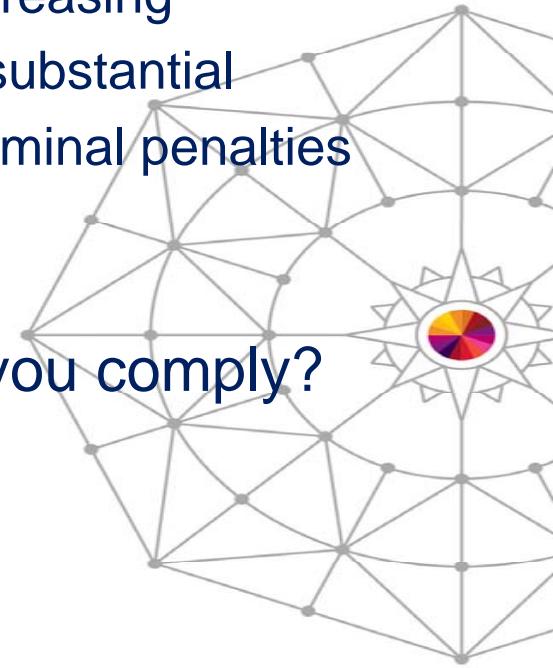
- Cost of Delayed Opportunity
  - Brittle systems mean change is hard, takes time
  - Competitors gain advantage
  - Profit lost forever



# Compliance Costs



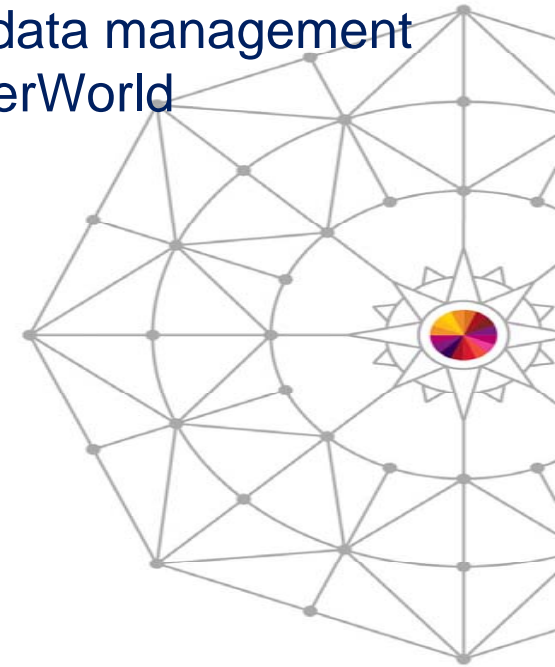
- Government Regulation
  - Regulation increasing
  - Fines can be substantial
  - Sometimes criminal penalties exist
- How fast can you comply?



# Do the math



- Intel reports saving \$6 for every one \$1 spent on meta data management – Source, ComputerWorld



# Social Security Administration & FSTAP

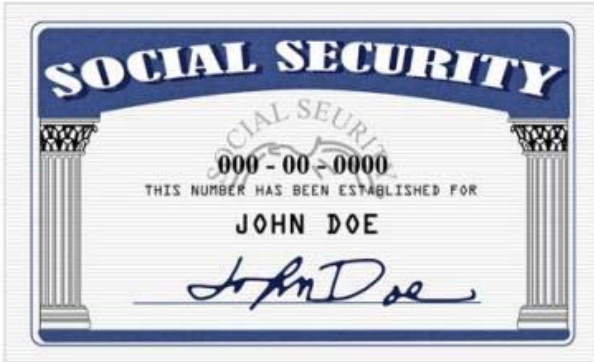
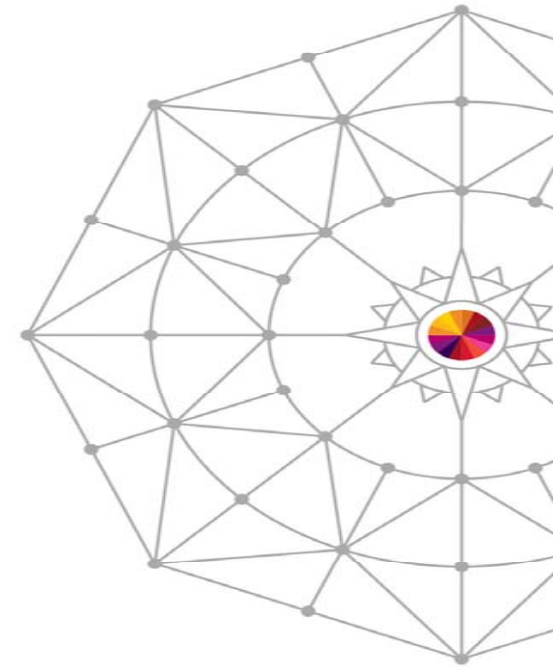


Image: LBNL

- 30
- 250 MM
- 250+
- 17+
- 93%





Rescue is at Hand

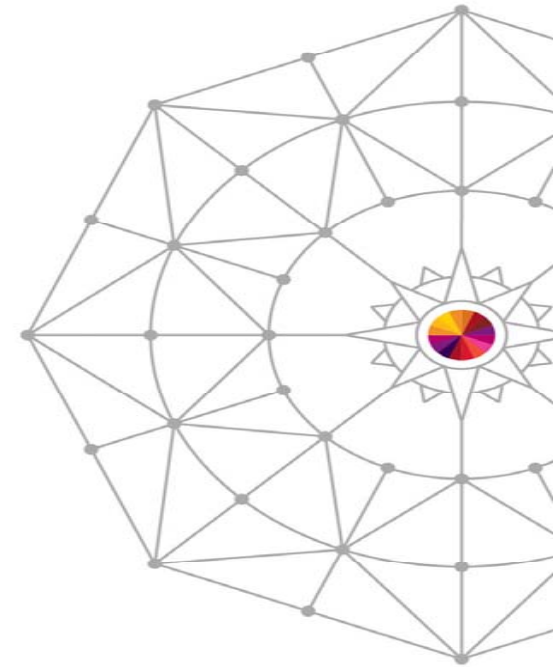
**15 months**



- Evaluate
- Analyze
- Remediate

**100%**

***Automated***

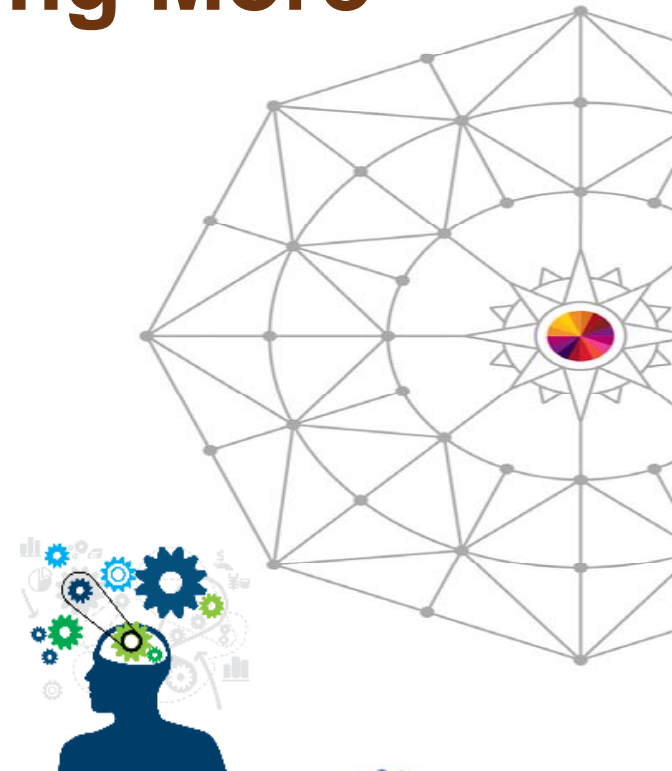




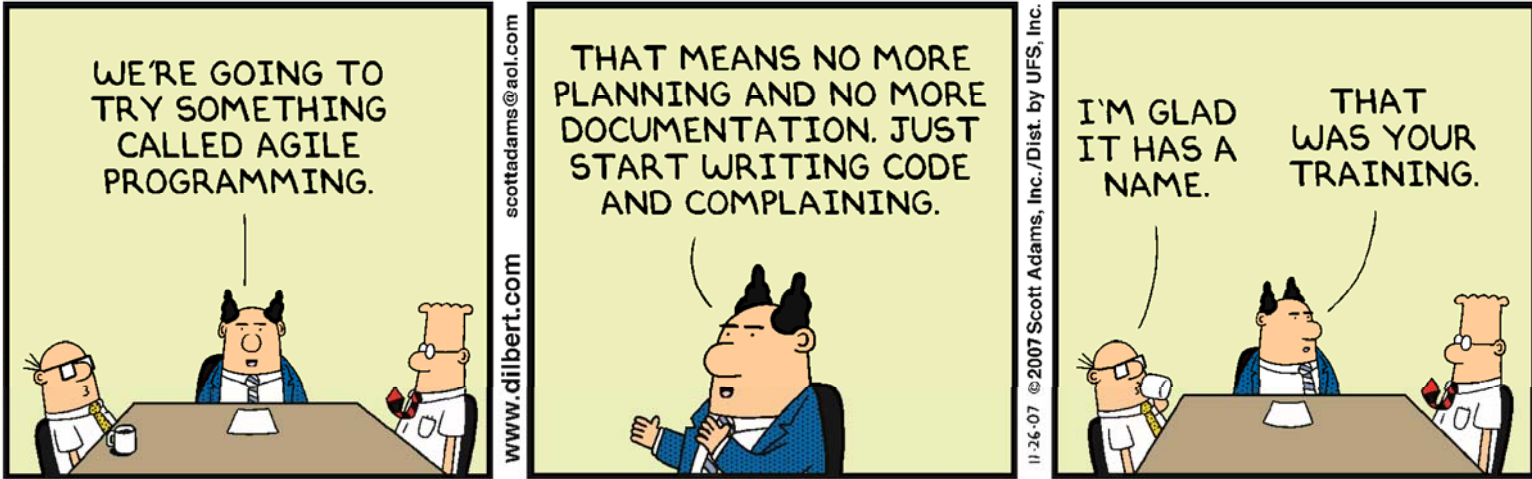


Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)

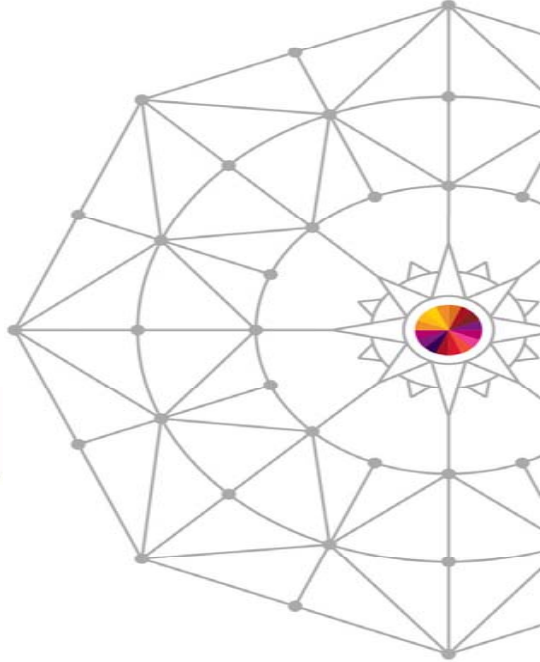
# Becoming More Agile



# The easy way to add agility to business software!



• DILBERT © 2007 Scott Adams. Used By permission of UNIVERSAL UCLICK. All rights reserved.



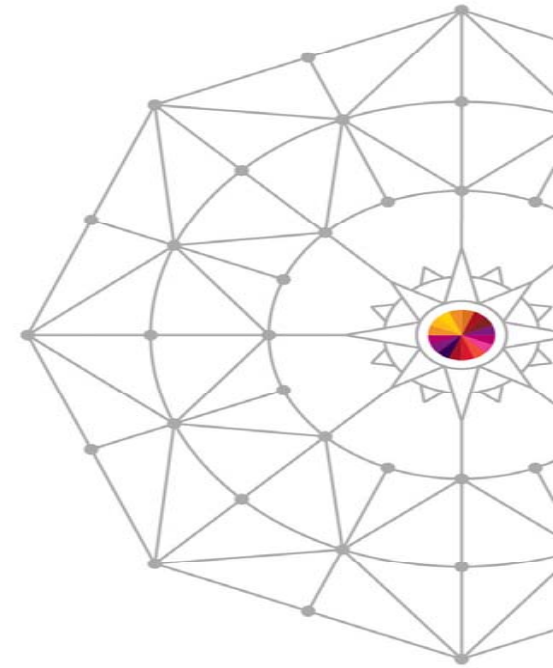
Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)

## Find your internal constraints

**People:** Lack of skilled people limits the system. Mental models held by people can cause behavior that becomes a constraint.

**Policy:** A written or unwritten policy prevents the system from making more.

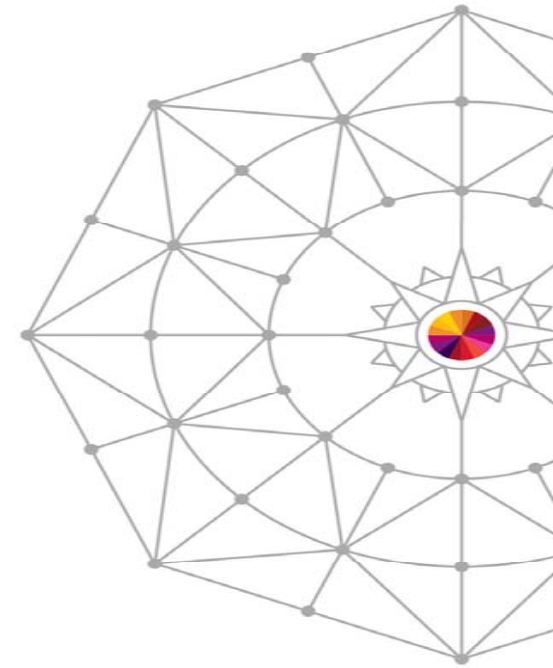
**Equipment:** The way equipment (i.e. software) is currently used limits the ability of the system to produce more salable goods/services.



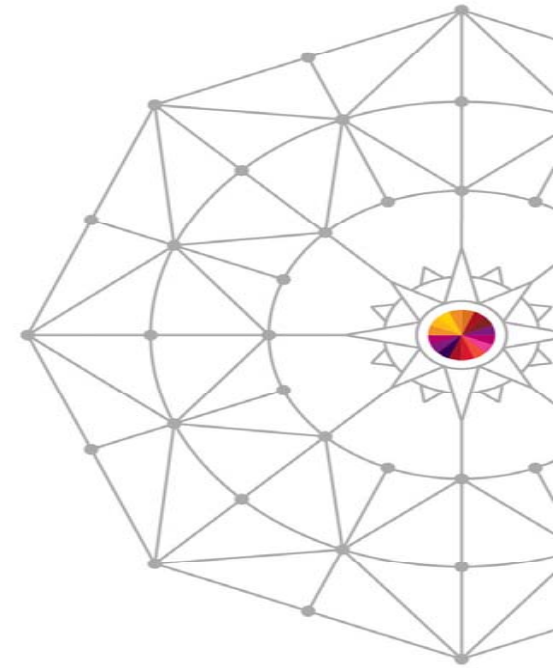
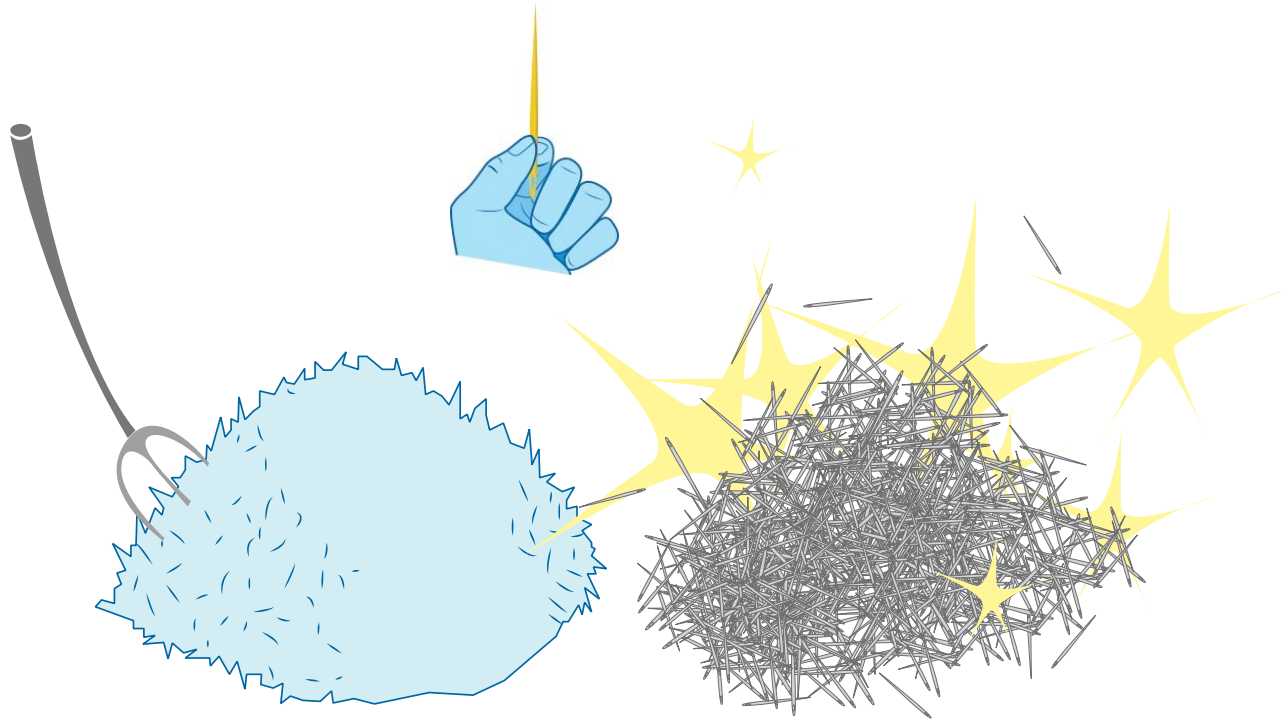
## How can the problem be solved?

- Rewrite / Replace
  - Large capital investment
  - Fact: most rewrite projects fail
- Status Quo
  - You know what happens here
- Or...

**Improve what you have already invested in**



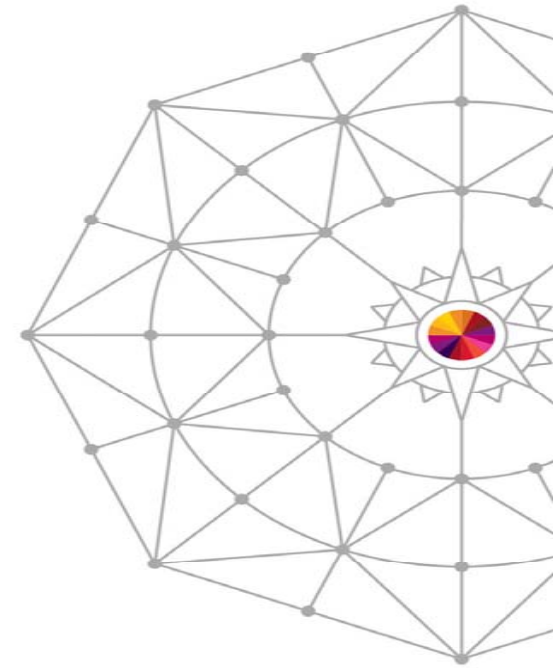
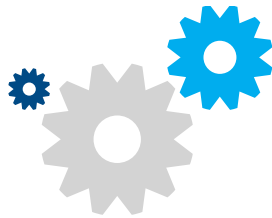
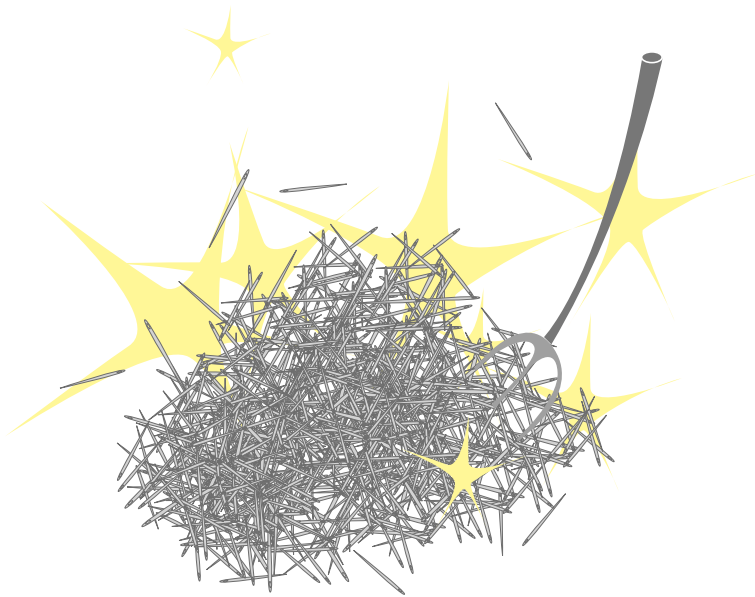
# Barriers to Agility – Sheer Mass of Code



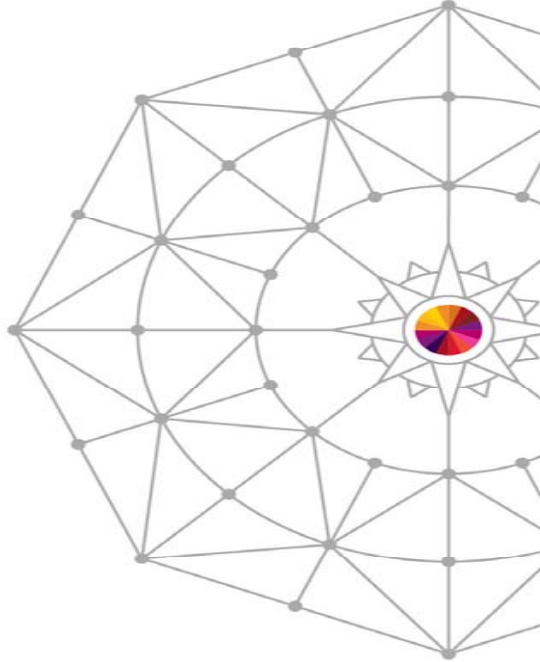
Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)



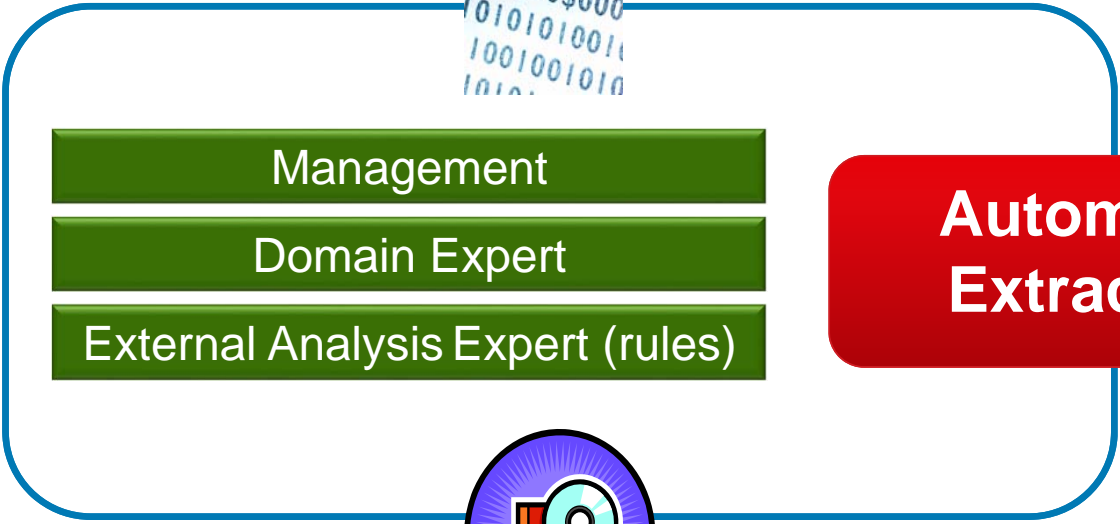
# What if we had a machine to do this?







# Enterprise Metadata Management Framework

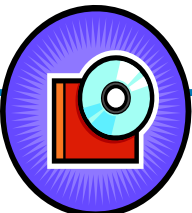


Management

Domain Expert

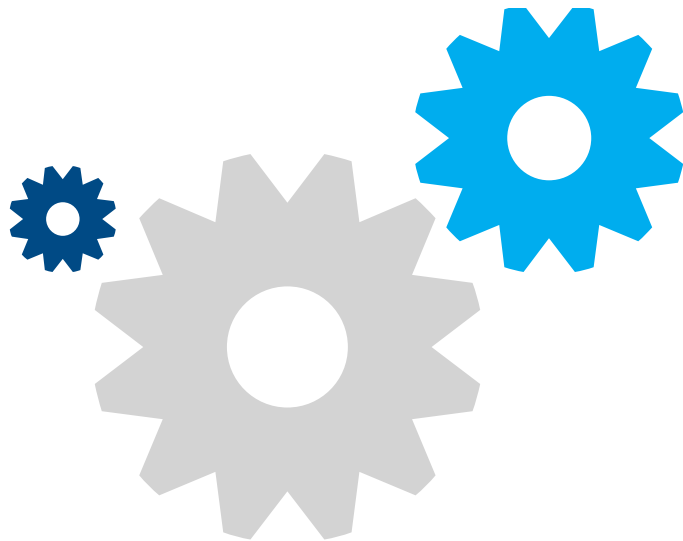
External Analysis Expert (rules)

**Automatically  
Extract Facts**

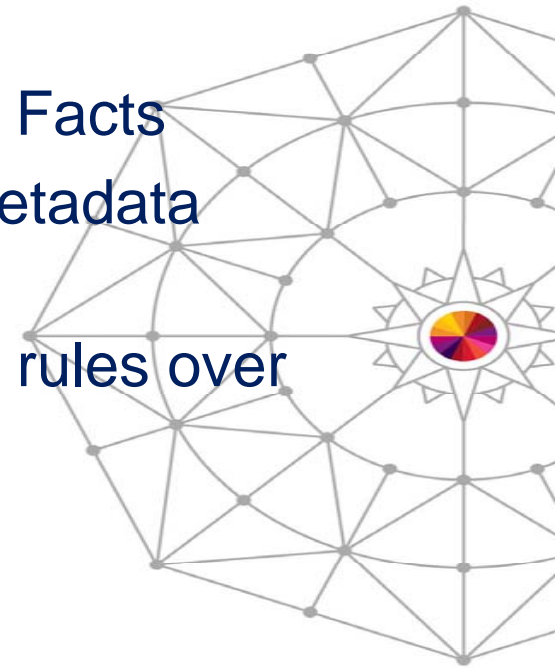


Metadata repository

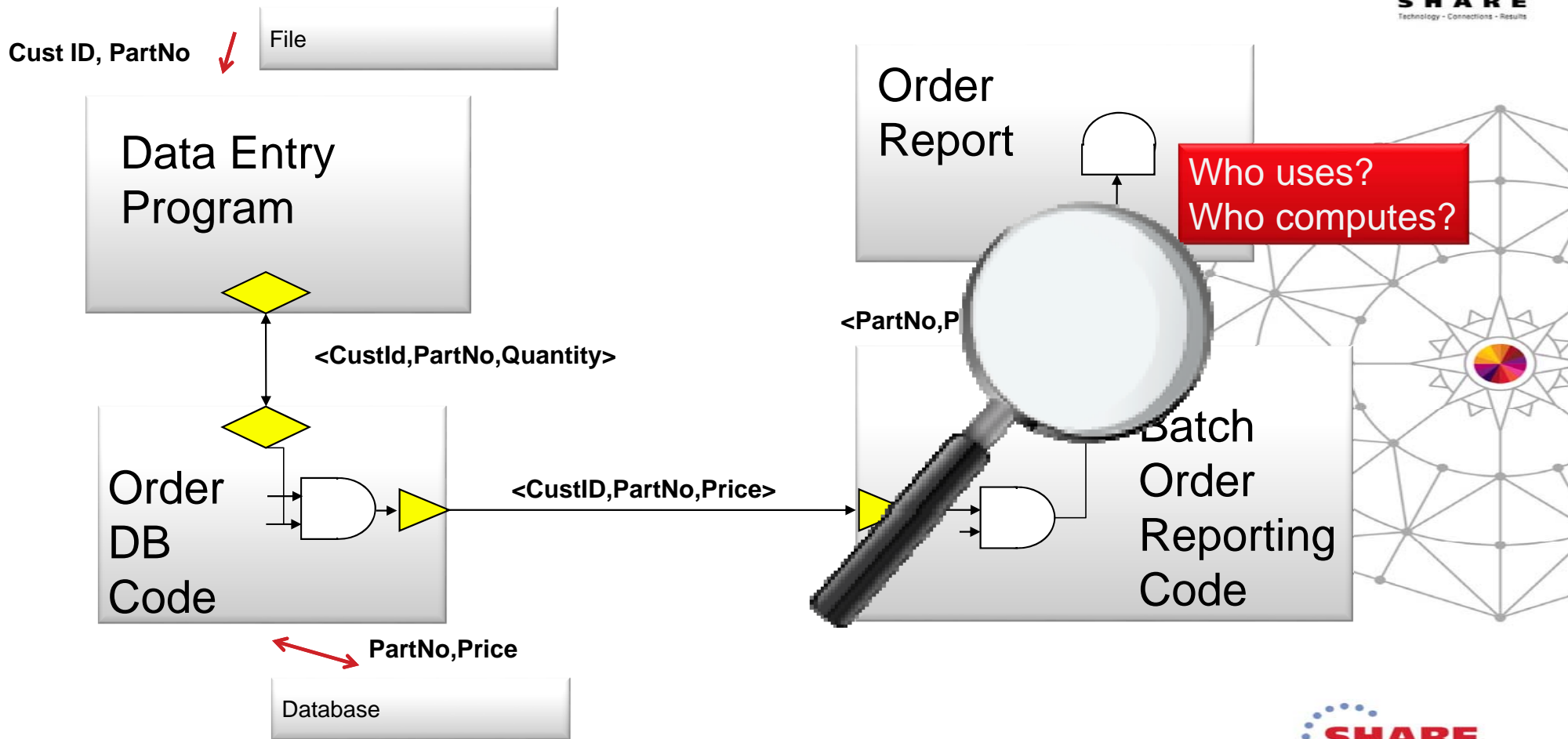
# Automated Software Discovery



- Read your code bases and schemas
- Extract Relevant Facts
- Populate your metadata repository
- Adapt and refine rules over time



# Automatically Discovering Connectivity / Flow



# MetaData Repositories

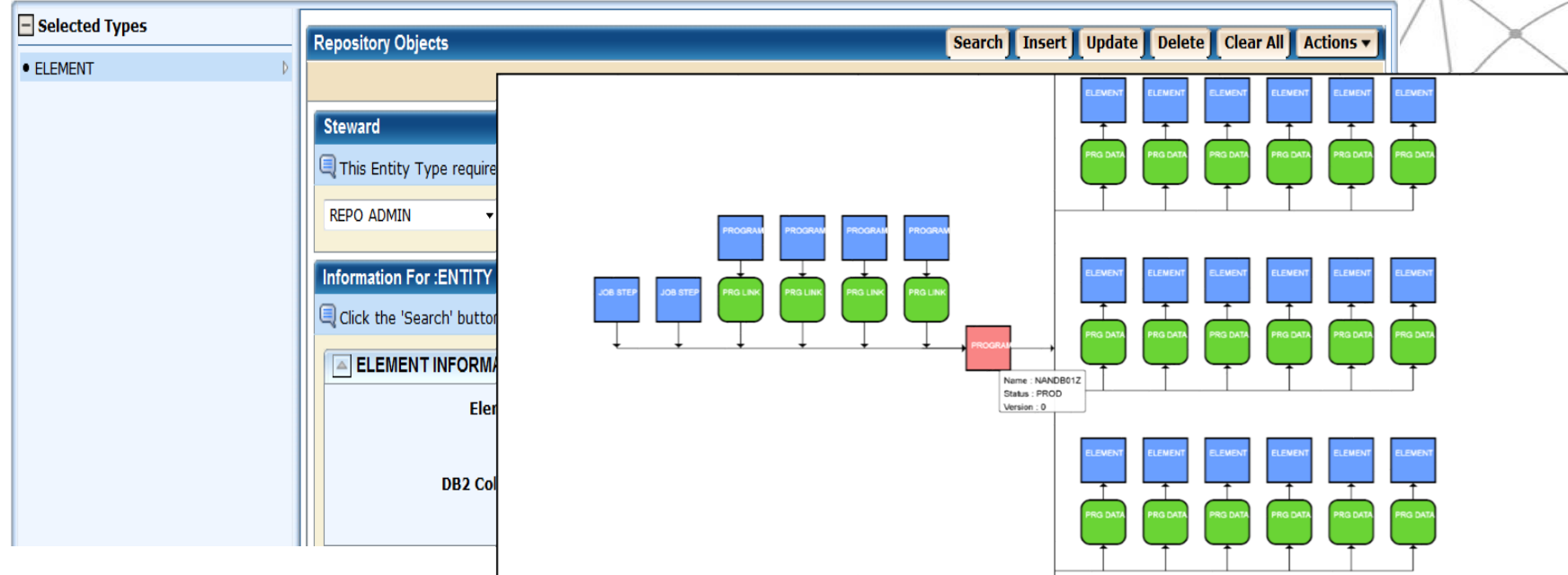
SHARE  
Technology - Connections - Results

ca.® Repository For z/OS Webstation Option

Logged in as: BOSRY01 (Log Out)

Dialog > DB2 (Types) > ELEMENT

Home Finder Categories Global Reports Repository Approval Profile Administration



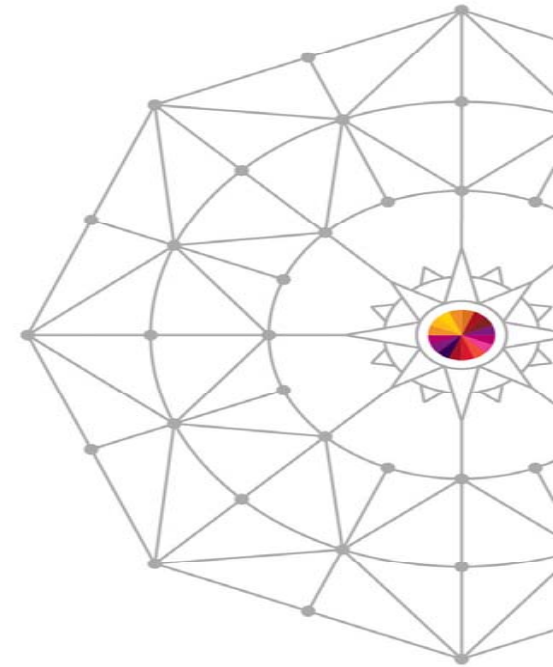
Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)

SHARE  
in Anaheim

# Code Quality Metrics

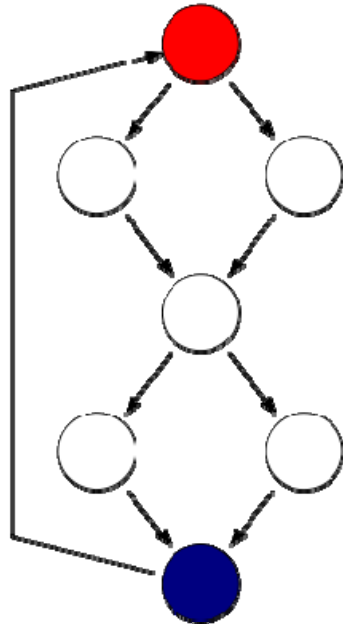


- Assess the quality of the code base
- Track and roll up
- Typical Metrics
  - Halstead/McCabe
  - SEI / Maintainability

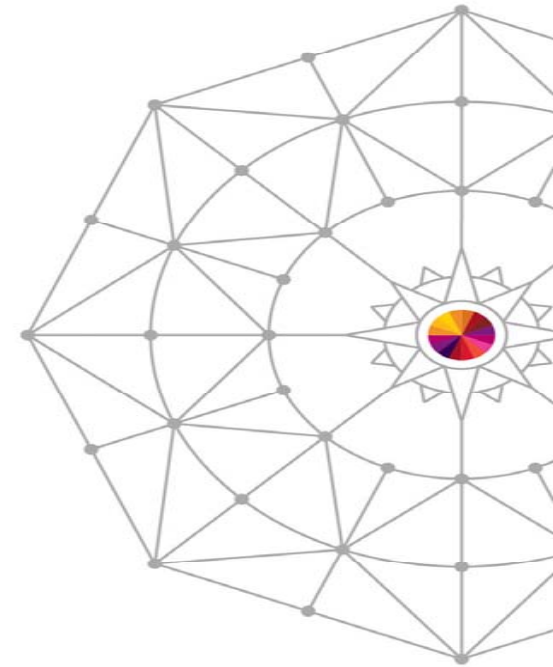


# Cyclomatic Complexity Metrics

```
if( c1() )  
f1(); else  
f2();  
  if( c2() )  
f3(); else  
f4();
```



The cyclomatic complexity of the program is 3 (as the strongly connected graph for the program contains 9 edges, 7 nodes and 1 connected component)  $(9 - 7 + 1)$ .



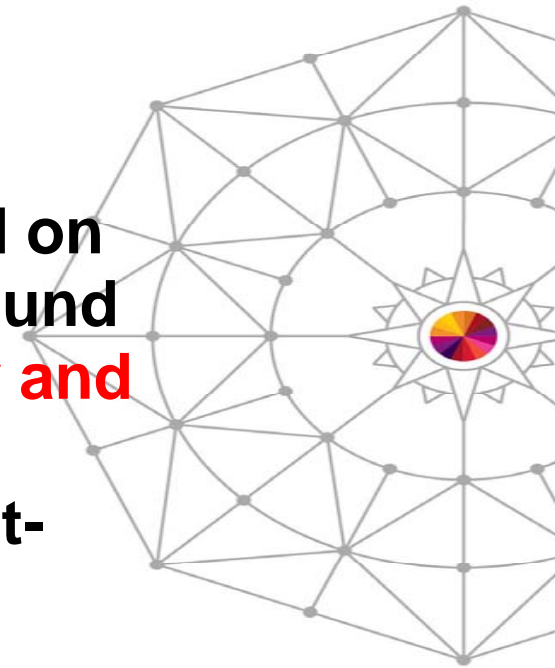
## Good Score < 10-15



## Complexity Metrics

[http://en.wikipedia.org/wiki/Cyclomatic\\_complexity](http://en.wikipedia.org/wiki/Cyclomatic_complexity)

**Enerjy** analyzed classes of open-source Java applications and divided them into two sets based on how commonly faults were found in them. They found **strong correlation** between **cyclomatic complexity** and **their faultiness**, with classes with a combined complexity of 11 having a probability of being fault-prone of just 0.28, rising to 0.98 for classes with a complexity of 74.



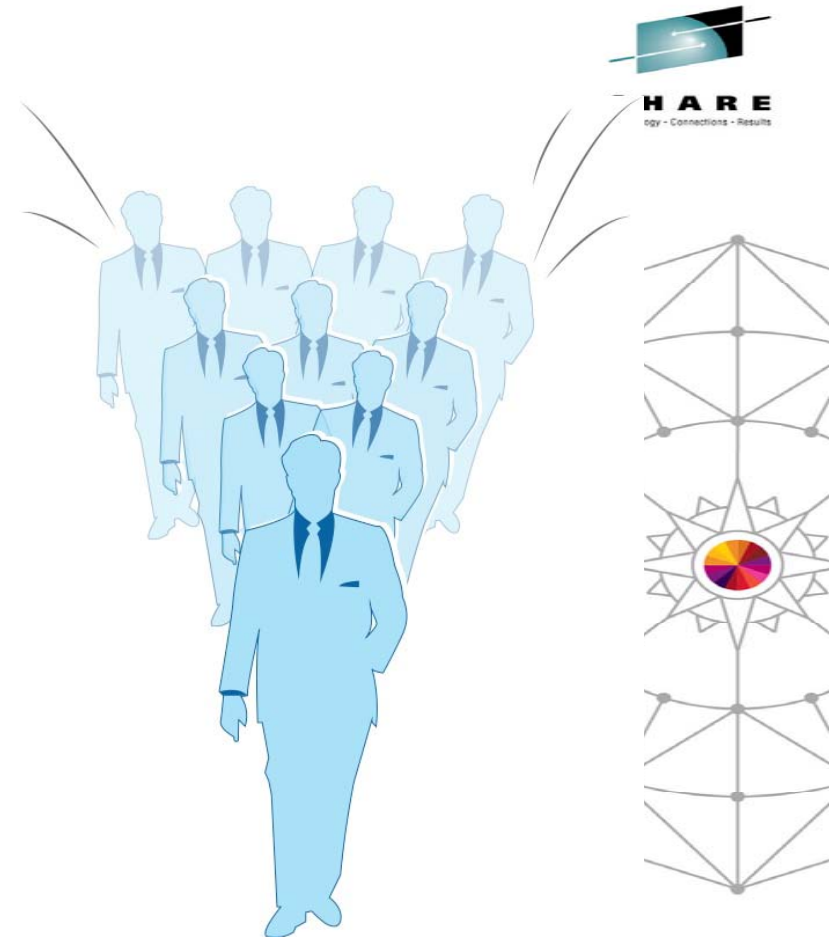
# Example Analysis

Source Lines	Code Lines	Comment Lines	Blank Lines	Cyclomatic Complexity	Halstead Complexity	Filename
1884	1884	0	0	497	4595350.5	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional KPI.Divisional KPI Report.Print Divisional KPI Report.TXT
4618	4616	2	0	214	1.34E+07	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional KPI.Divisional KPI Report.Print Divisional KPI Report.TXT
2487	2479	7	2	204	4713399.5	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional KPI.Divisional KPI Report.Print Divisional KPI Report.TXT
3161	3158	3	0	199	7425025	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Customer Service Data.Delivery Information.Maintenance suite.Change user interface.TXT
1173	1172	1	0	184	1837039.2	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional KPI.Divisional KPI Report.Print Divisional KPI Report.TXT
1430	1414	29	5	173	4855147.5	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional KPI.Divisional KPI Report.Print Divisional KPI Report.TXT
2247	2241	6	0	170	3829034.5	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Customer Service Data.Delivery Information.Maintenance suite.Change user interface.TXT
2747	2732	14	1	166	8507429	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Customer Service Data.Delivery Information.Maintenance suite.Change user interface.TXT
1318	1316	2	0	165	2103735	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Fire Invoice.UI.DetailMaint.Change Service Visit Item.TXT

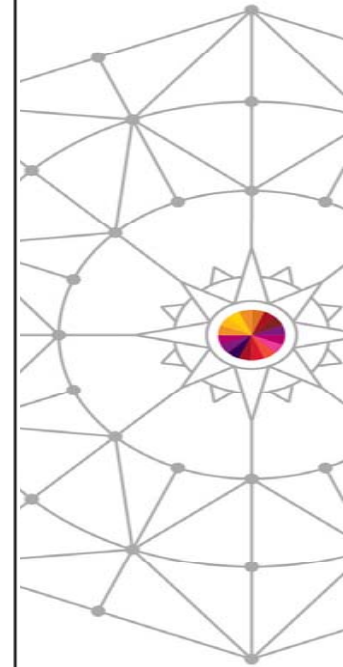
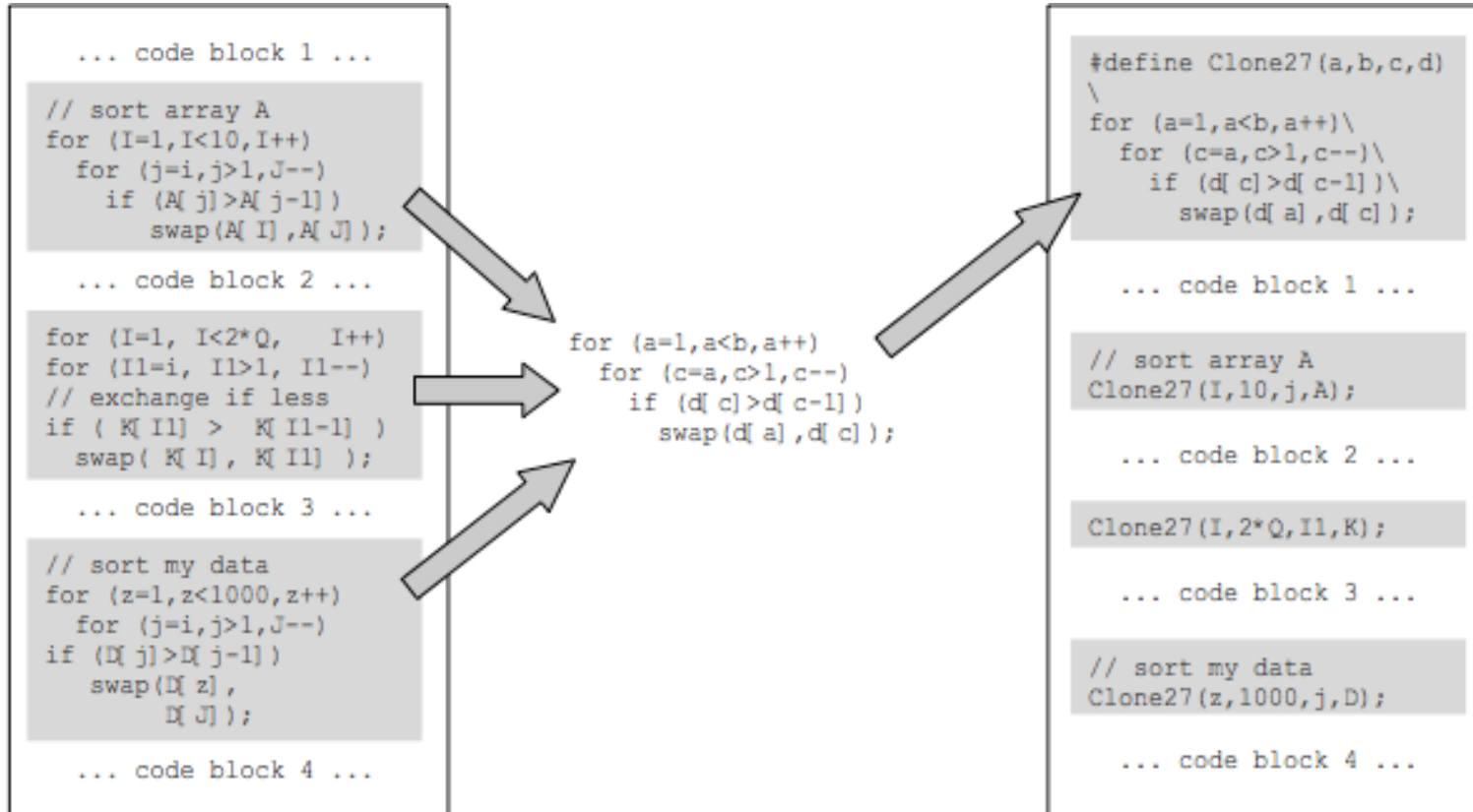
Analyze the programs costing the most money to maintain

## Cloned Code

- Clones cost money
  - \$1/year to own code
  - Chances of error
  - Multiple Maintenance
- 15%-25% typically cloned
- Can be difficult – not just string matching



# Detecting / Remediating Clones

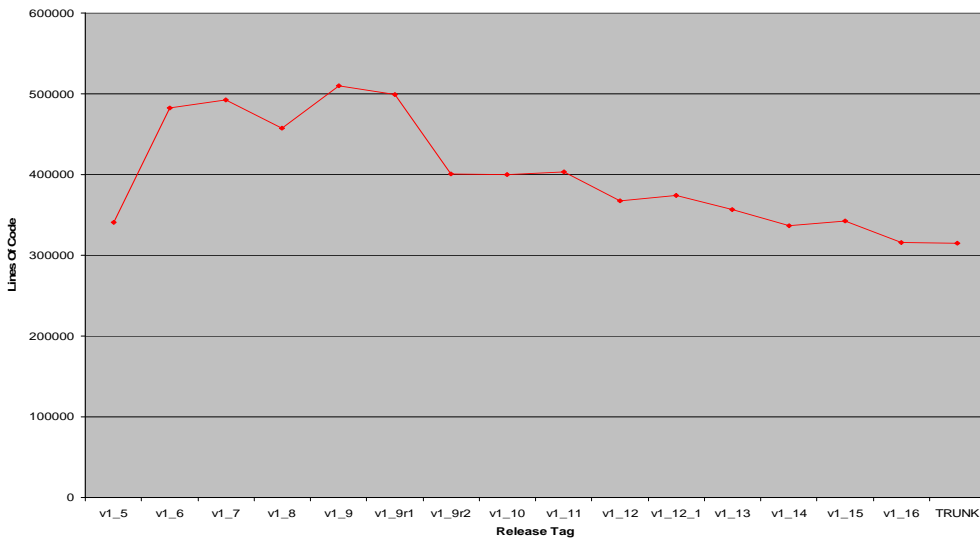


# Example – CRM System

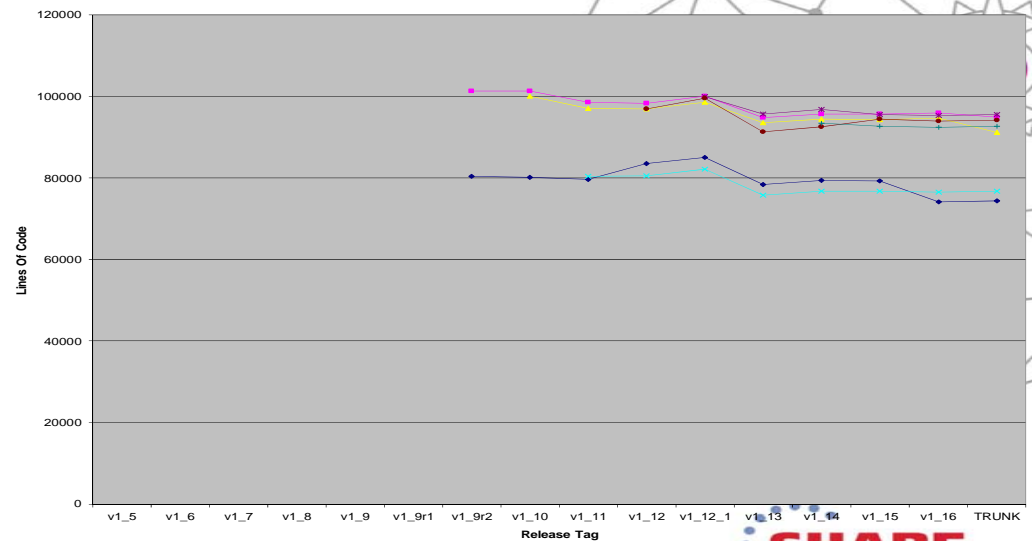
- *Java Application: Core + 6 Customer variants*
- *Active clone removal by development team*
- *Total SLOC reduction over time: ~40%*

*“Most of the drop in the source code graph (core) is due to clone detection removal.”*  
**Customer Quote**

Core Code Trend Analysis



Custom Code Trend



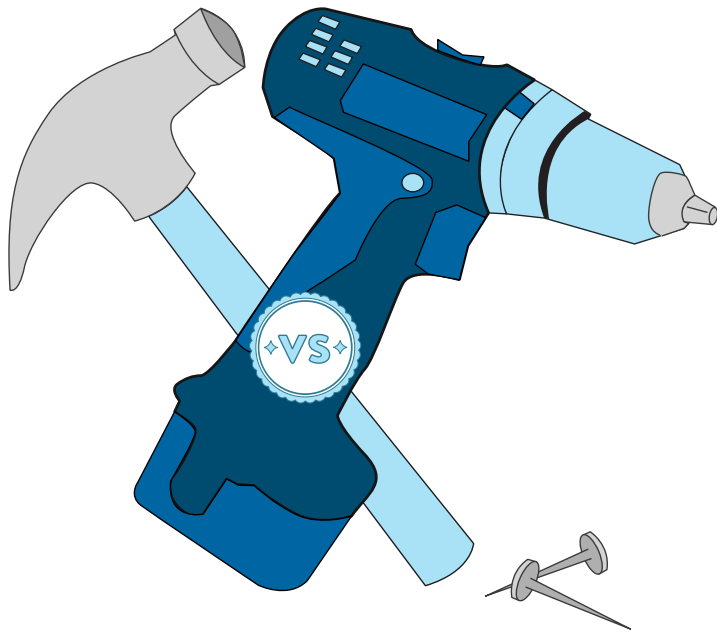
# Solving The Problem - Smarter

What do you need to look for?





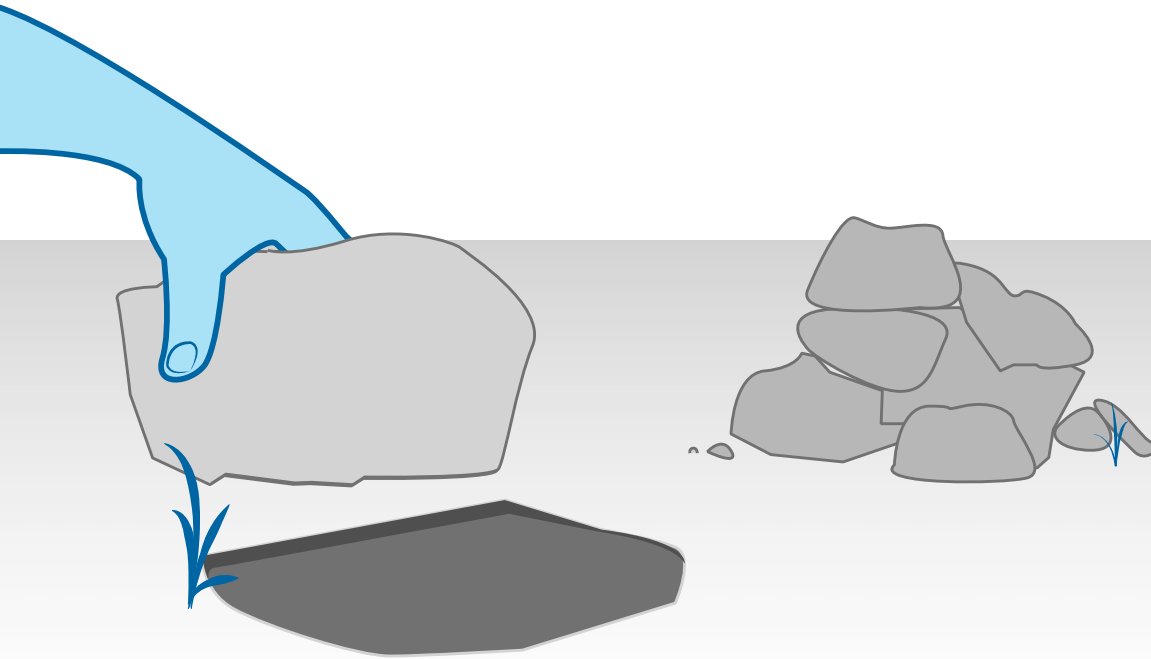
# Automation Required



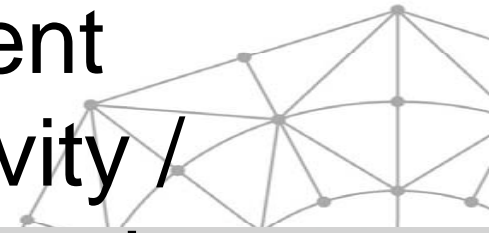
- Impossible for humans to manually analyze large software systems
- Must contain enough of the code base to make a difference



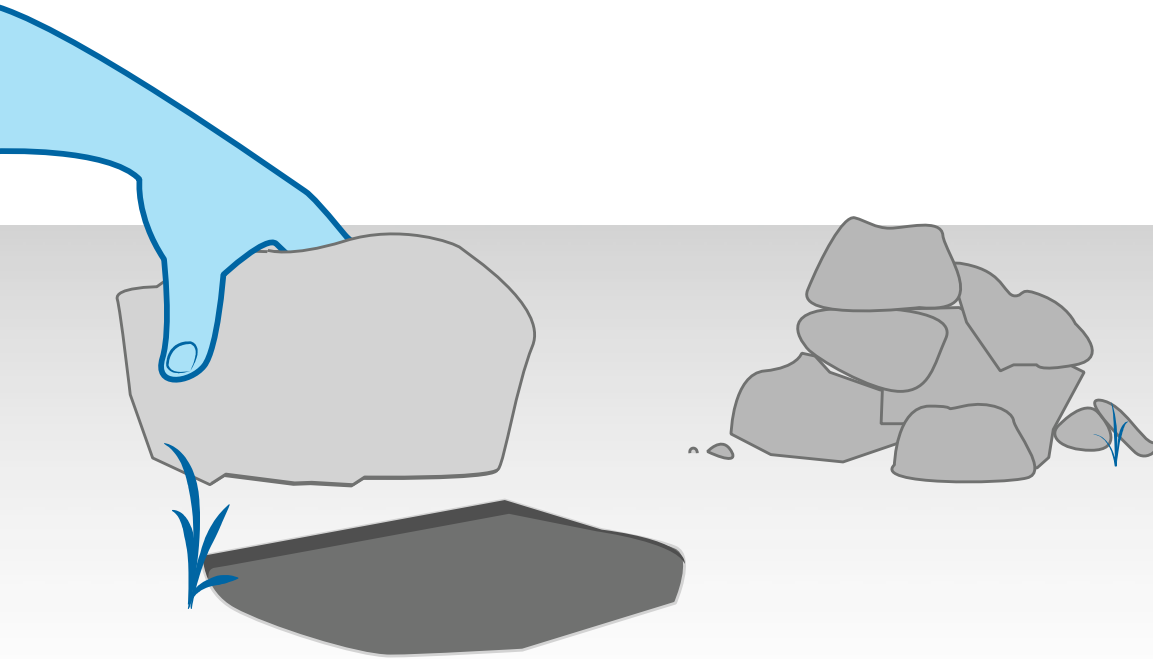
## Does the solution cover the basics?



- Component Connectivity / Where Used
- Metrics
- Clones
- Test Coverage

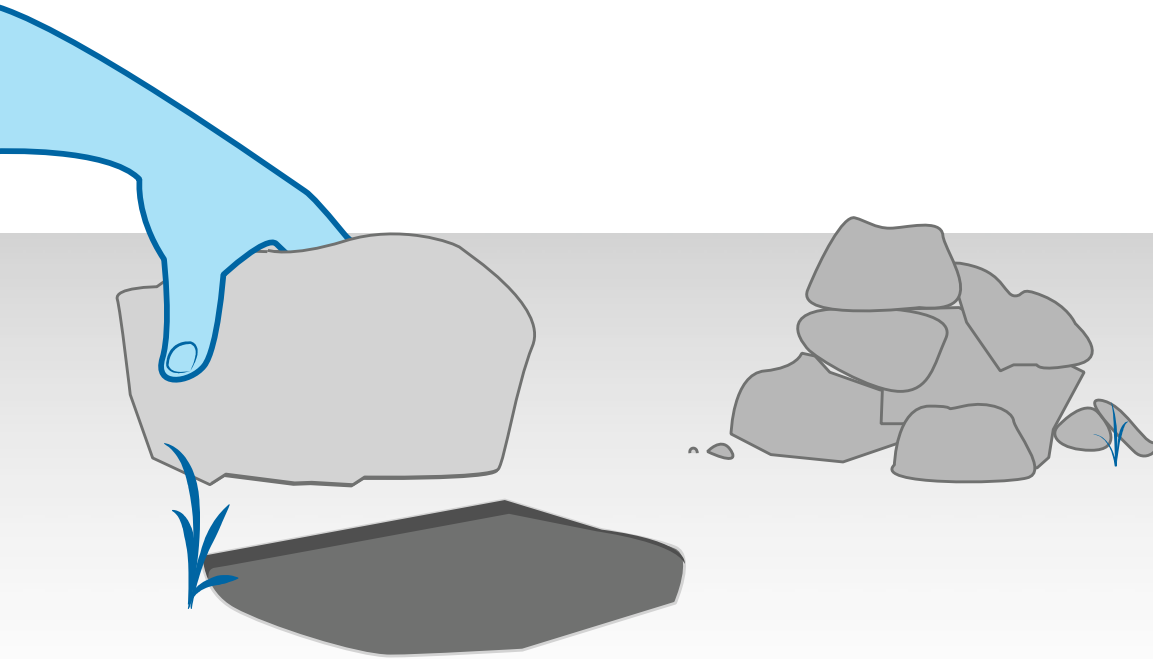


## Is the solution enterprise grade?



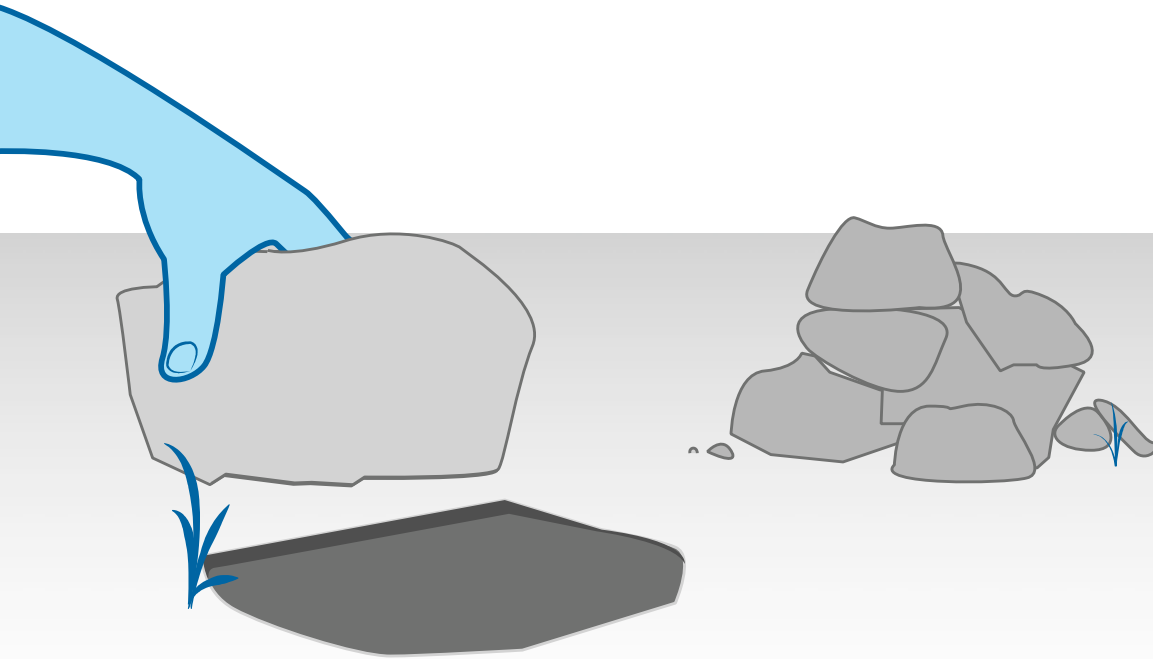
- Link to SCM's like Endeavor, PTC
- Scan code base in daily window
- Handle millions of lines and resulting data in the EMR

# Does the solution provide compiler-accurate results?



- Dynamic calls
- Dynamic SQL
- Flow Analysis
- False positives waste significant time

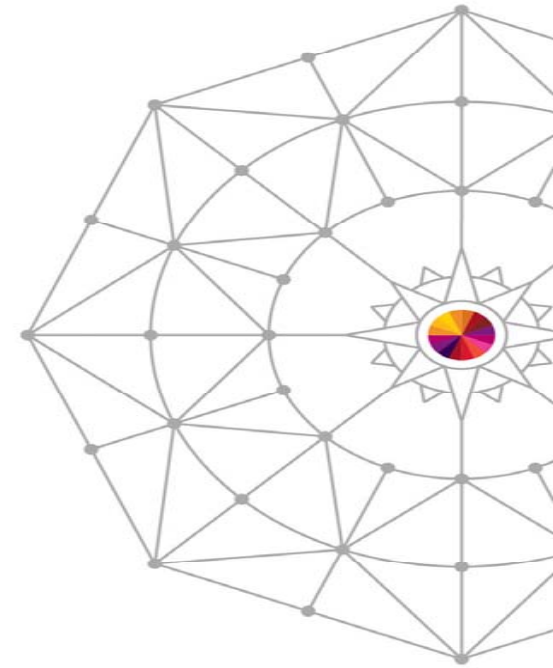
## Does the solution cover your code base?



- Wide Range of Domains
- Enterprise: COBOL, JCL, ALM
- OO: Java, C#
- Web: JavaScript, HTML, PHP

# Summary Video

[YouTube](#)



Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)





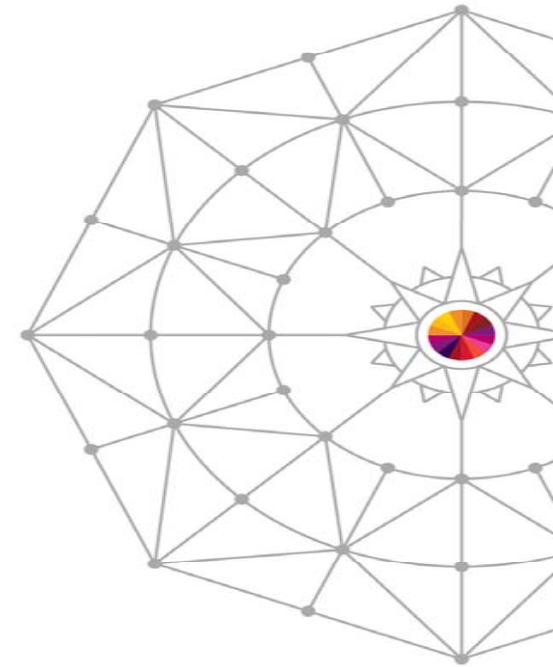
## Summary

- **Lack of agility can be costing your company**
  - Outdated Documentation
  - Older Architectures

- **Unlock your Code**
  - Metadata repositories
  - Automated Discovery
  - Code Visualization
  - Quality Metrics
  - Test Coverage
  - Clone Remediation



# Questions / Feedback



Complete your session evaluations online at [www.SHARE.org/Anaheim-Eval](http://www.SHARE.org/Anaheim-Eval)



So next week...



# Get more Agile and avoid Virtual Pratfalls!



[john.rhodes@cmfirstgroup.com](mailto:john.rhodes@cmfirstgroup.com)