Paying Down your Technical Debt

Booth 231

Technical Debt in a Nutshell

- ## Ward Cunningham Definition
  - Like financial debt, technical debt may have helped the company move faster
  - But like an overleveraged company, it can eventually paralyze operations

- ## Strategy for paying down the debt
  - Measurement and remediation with automation and metadata repositories

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

SHARE in Boston   SHARE   #SHAREorg   cm FIRST Rethink Modernization

Agenda

- The Technical Debt Story
- Definitions
- The Cost of Technical Debt

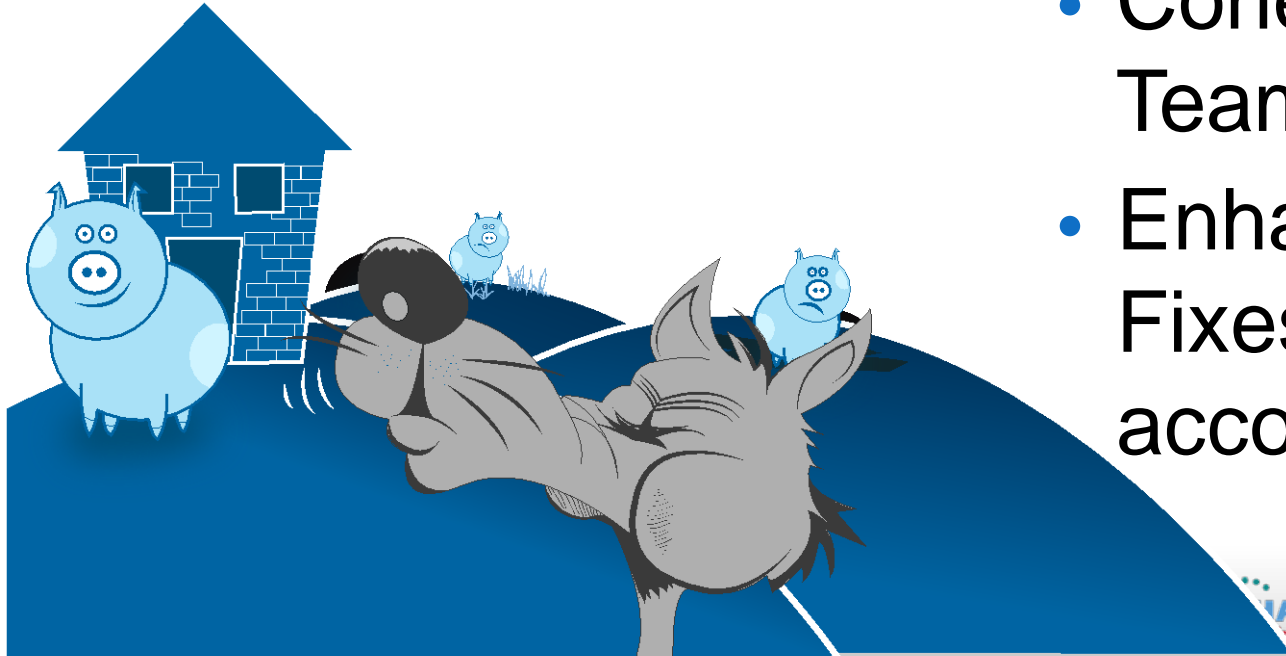- Solving the technical debt problem
- Solving it smarter
- Conclusions

The Technical Debt Story

You may be familiar with this ☺!

Initially There is no Problem

- Adherence to Design
- Cohesive Technical Team
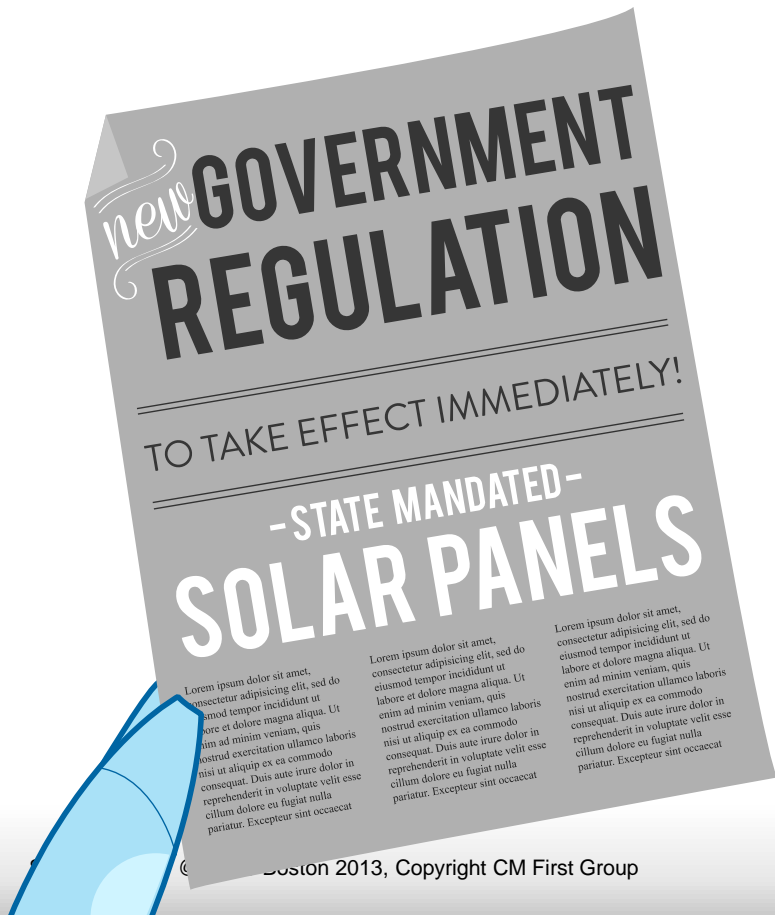- Enhancements and Fixes are readily accomplished
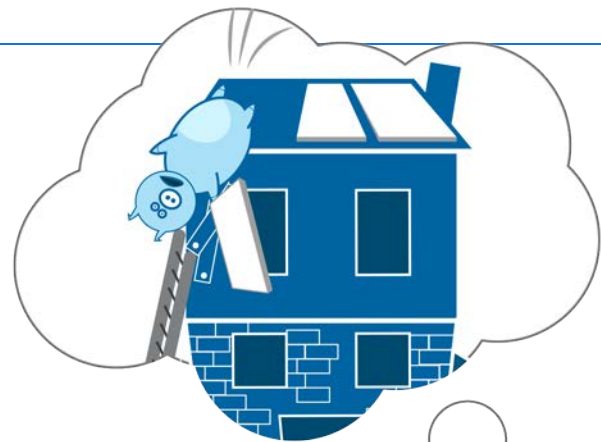
Technical Debt Grows over Time

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

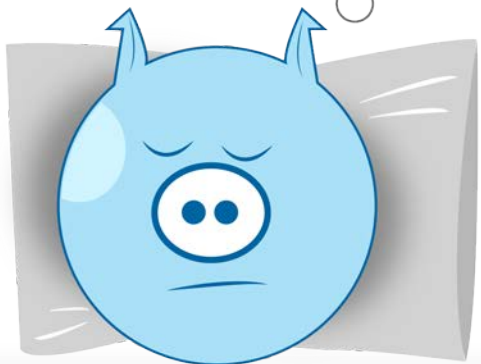# Until you reach a point where…

- 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!

The Cost of Technical Debt
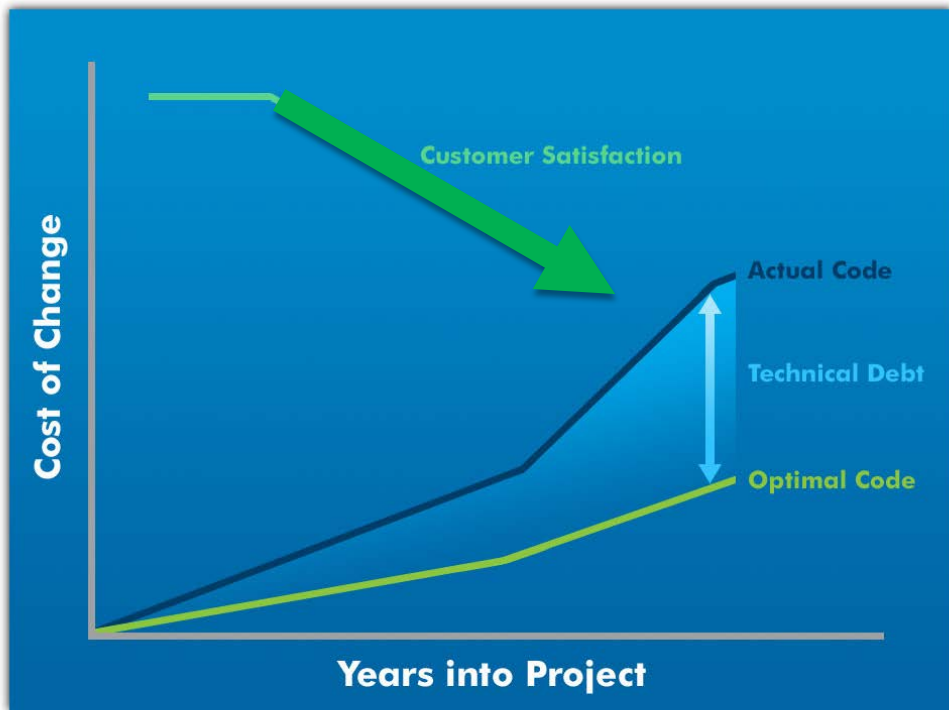
Technical Debt / Quality Cost
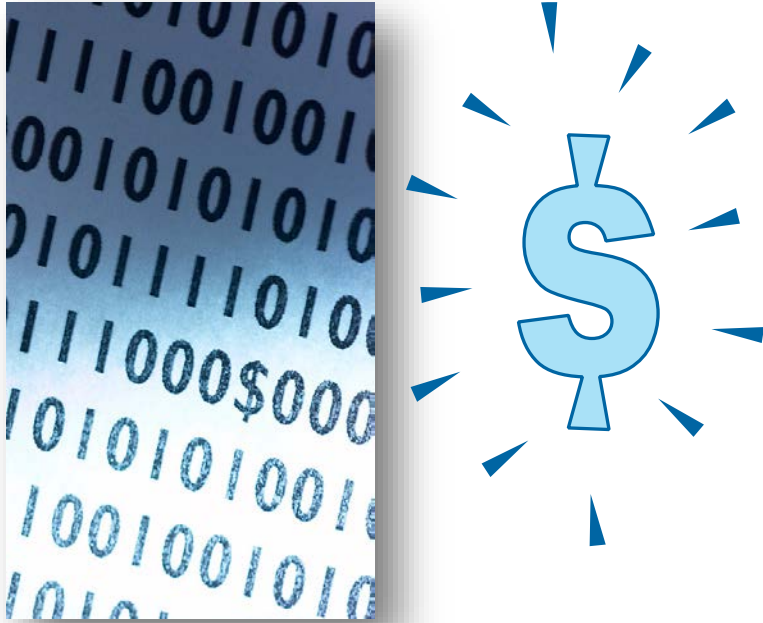

Fig. 1 - The Cost of Technical Debt

- 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
  - 60% of time spend finding/quantifying what to do
- Little time to spend on new business
- Outages, performance problems

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

#SHAREorg

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
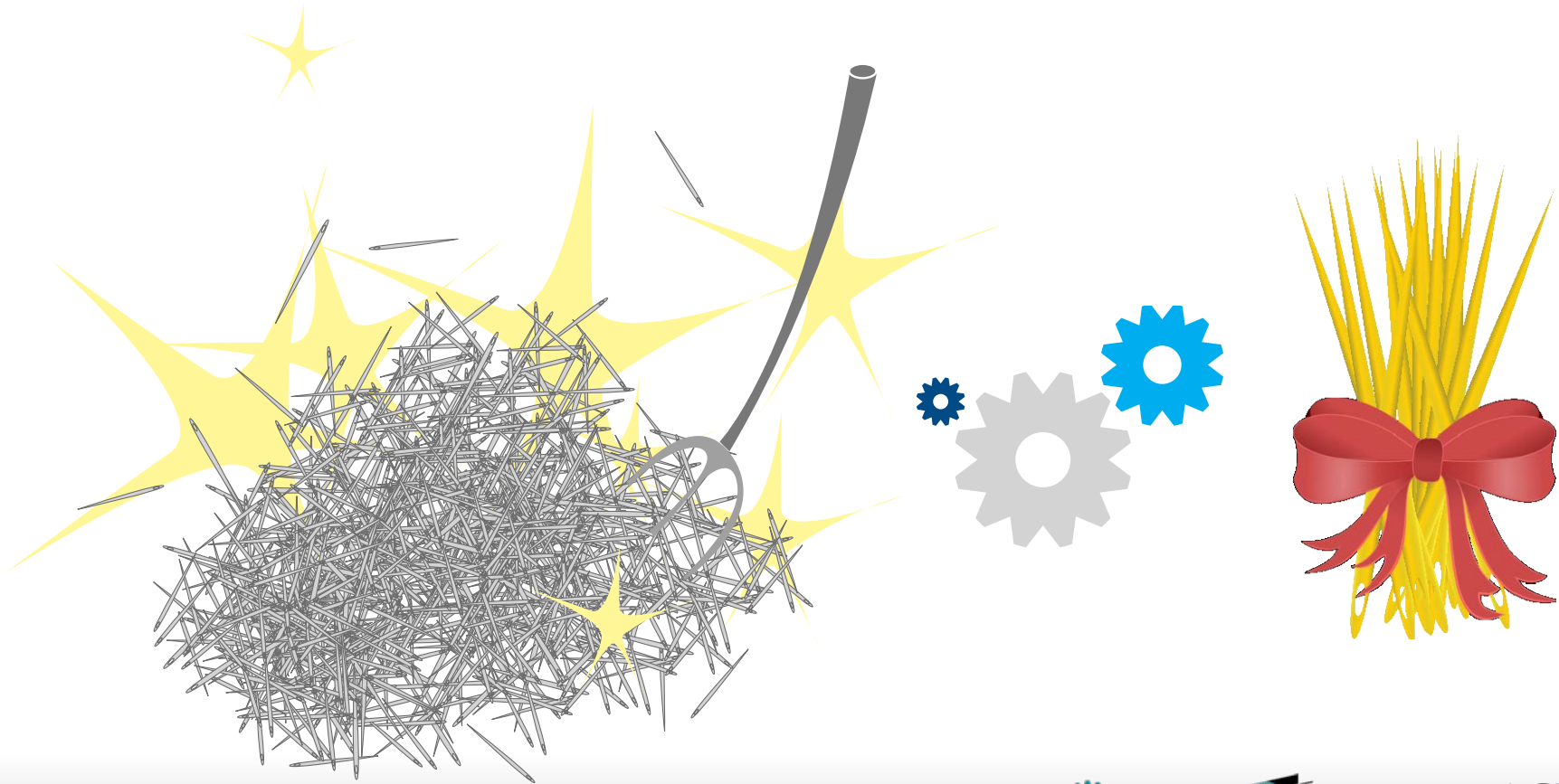
Solving the Problem

Paying Down your Technical Debt

How can the problem be solved?

- Rewrite / Replace
  - Fact: most rewrite projects fail
- Add more resources fixing code
  - Fact: cost are high, and are you making progress?
- Status Quo
  - You know what happens here
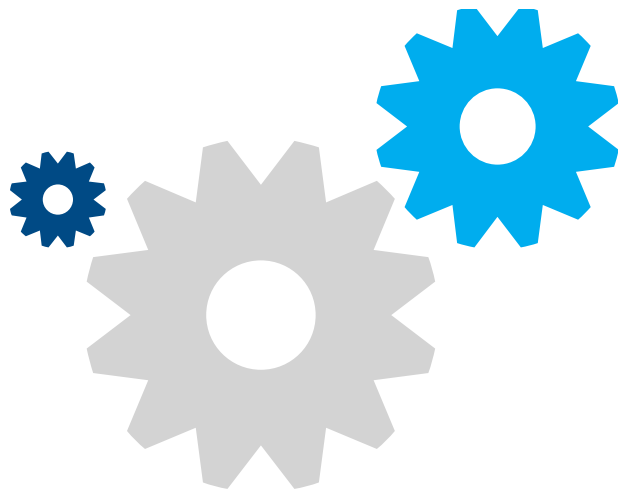- Or…

Why can't we move forward?
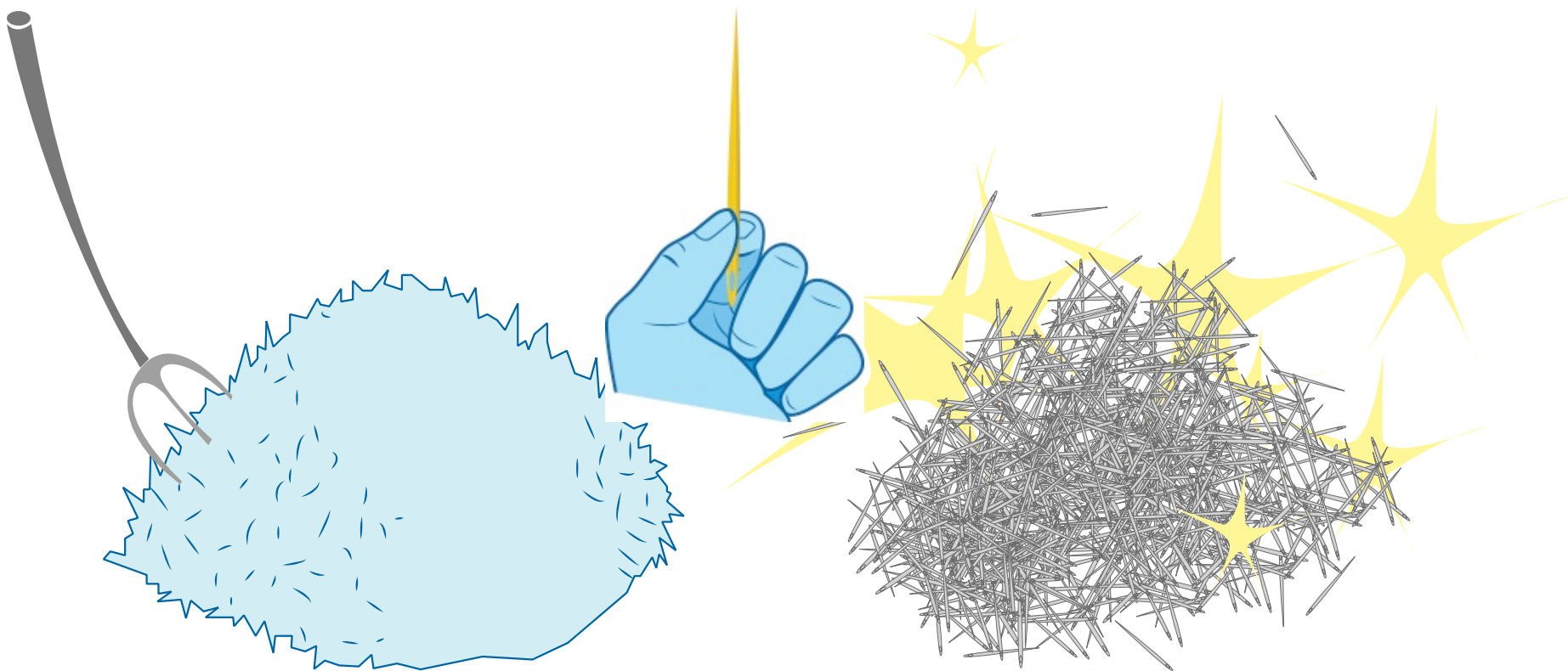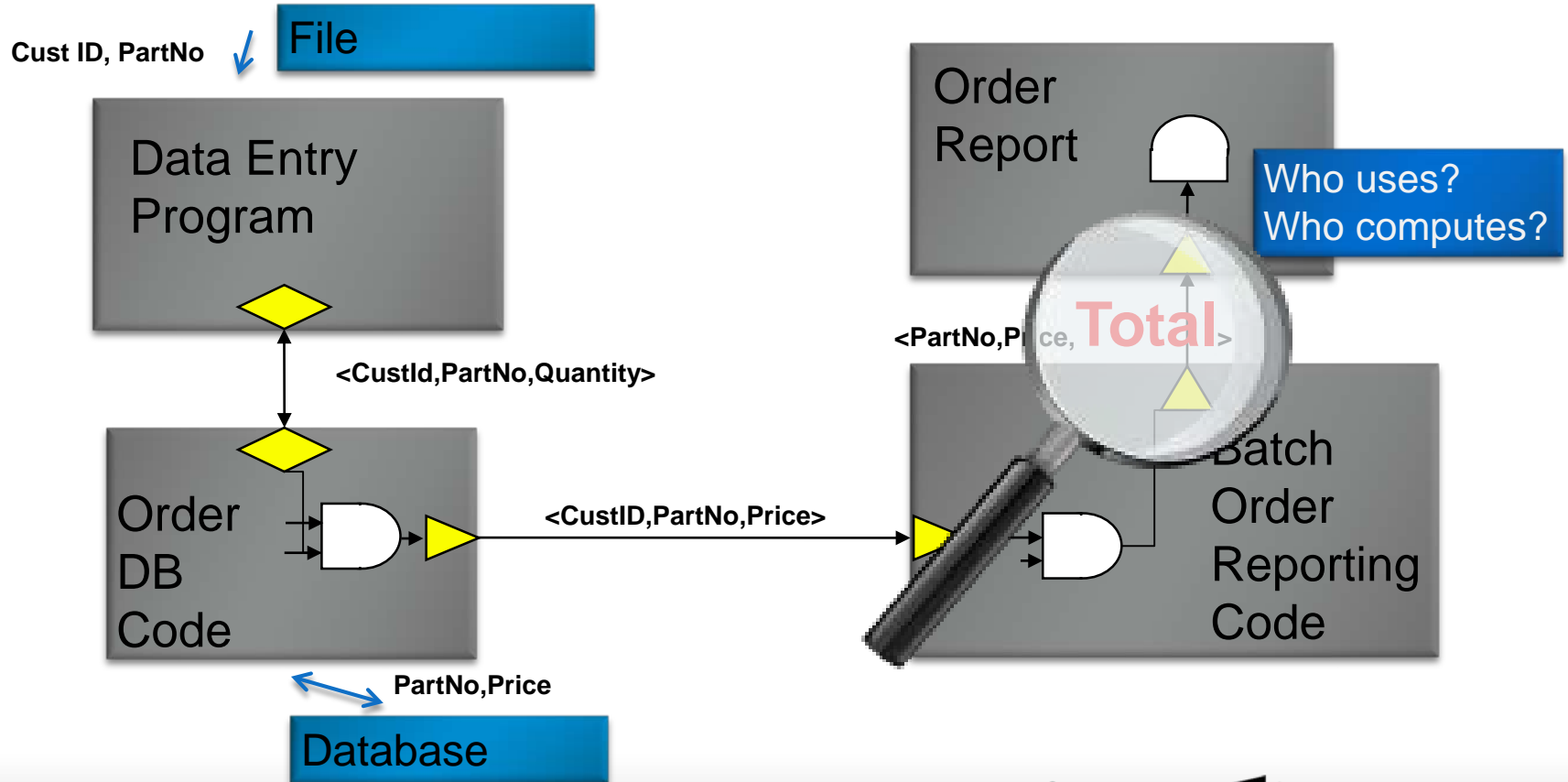
What if we had a machine to do this?

Debt Management Framework

Management

Domain Expert

External Analysis Expert (rules)

Automatically Extract Facts

Metadata repository
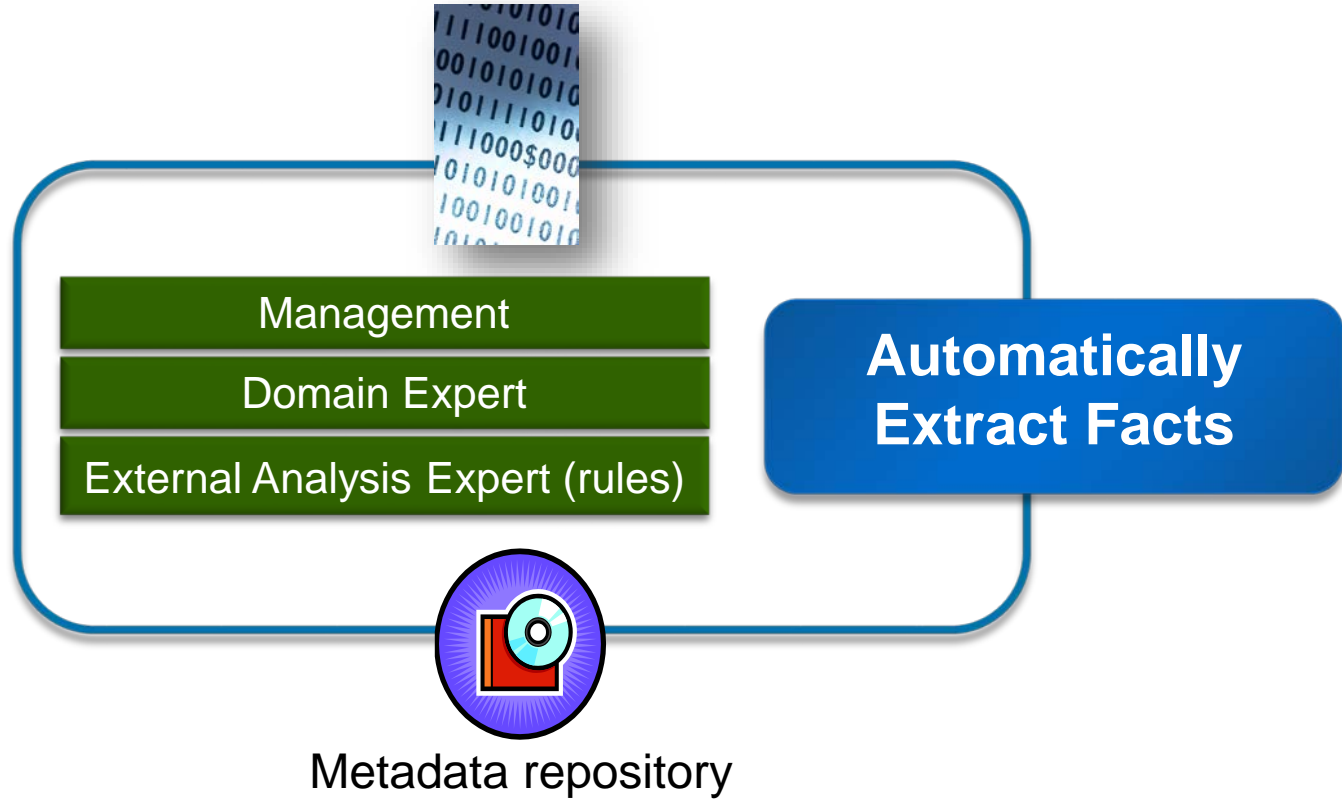
Automated Software Discovery

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

Automatically Discovering Connectivity / Flow

Cust ID, PartNo

File

Data Entry
Program

<CustId,PartNo,Quantity>

Order
DB
Code

<CustID,PartNo,Price>

PartNo,Price

Database

Order
Report

Who uses?
Who computes?

<PartNo,Price, Total>

Total

Batch
Order
Reporting
Code

MetaData Repositories

Code Quality Metrics

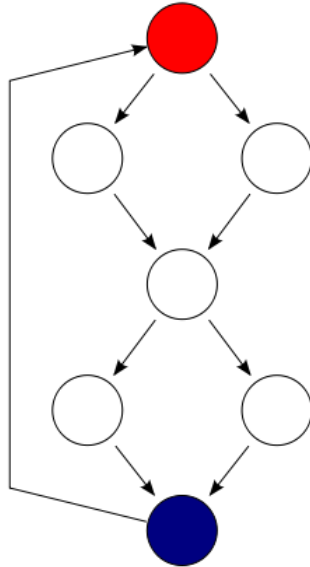- Assess the quality of the code base
- Track and roll up
- Typical Metrics
  - Halstead/McCabe
  - SEI / Maintainability

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

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.

#SHAREorg

CM FIRST
Rethink Modernization

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:/Documen... Exports/Exp...TXT
2487	2479	7	2	204	4713399.5	C:/Documen... Exports/Exp...
3161	3158	3	0	199	7425025	C:/Documen... Exports/Exp...nt Customer S...
1173	1172	1	0	184	1837039.2	C:/Documen... Exports/Exp...al KPI Report.TXT
1430	1414	29	5	173	4855147.5	C:/Documen... Exports/Exp...n Report.Prin...
2247	2241	6	0	170	3829034.5	C:/Documen... Exports/Exp...e user interface.TXT
2747	2732	14	1	166	8507429	C:/Documents and Settings/... Exports/ExportLargeProperty/Function/Function FirstAid Customer.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
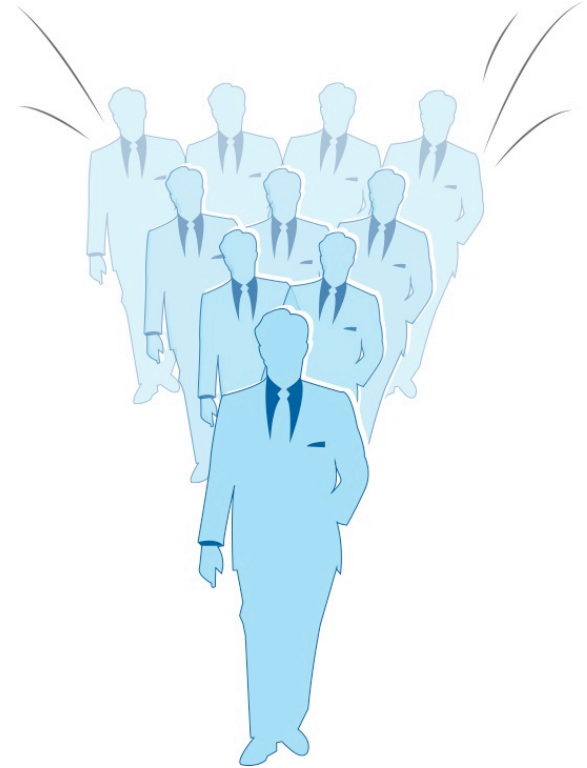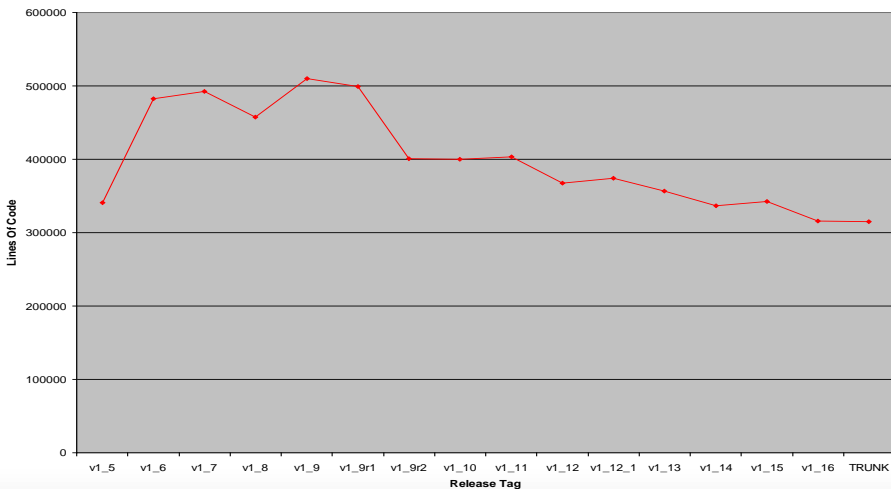
Detecting / Remediating Clones
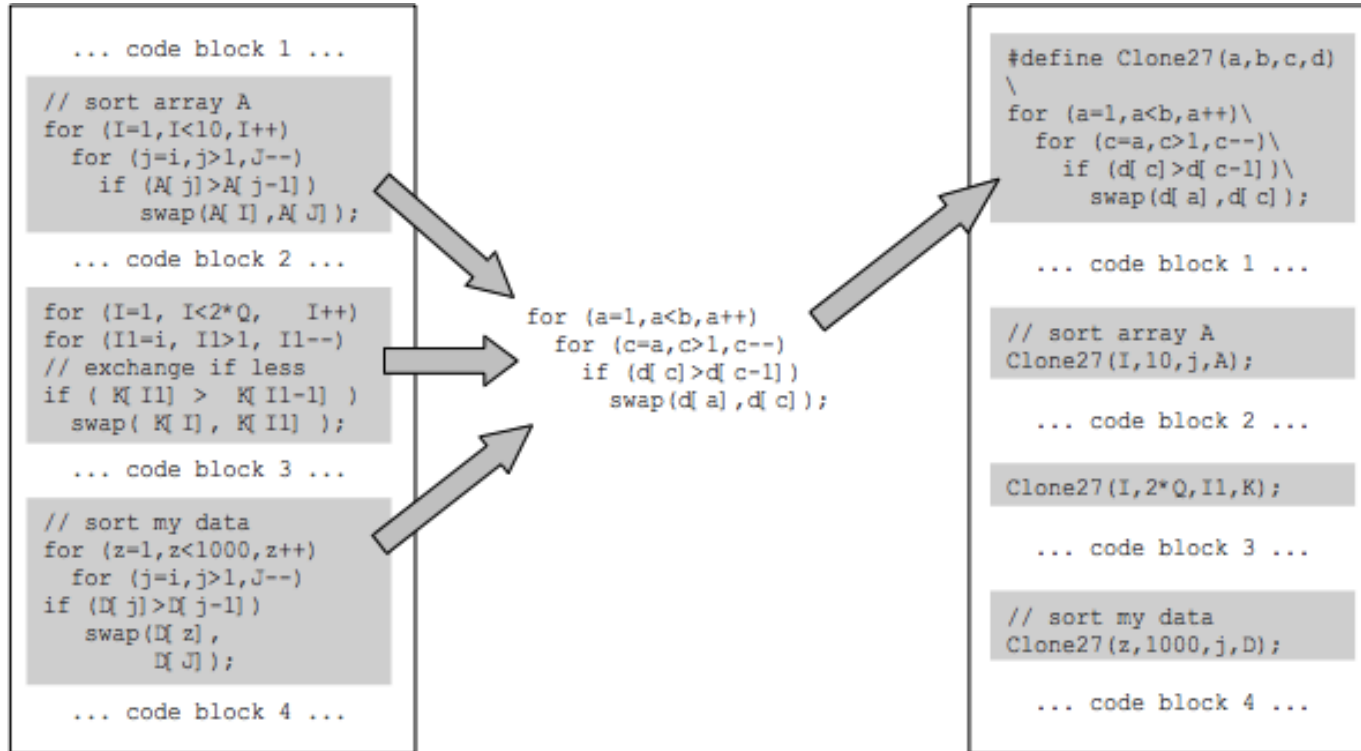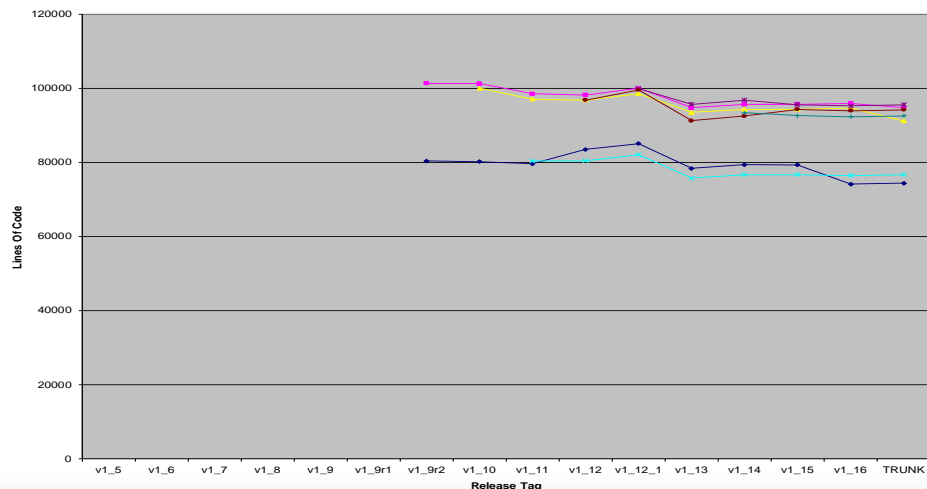
Example – CRM System

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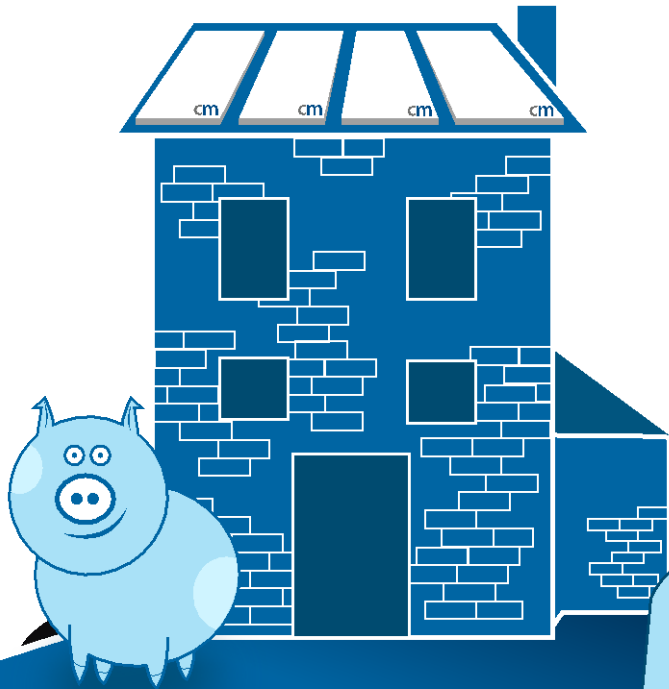- *Total SLOC reduction over time: ~40%*

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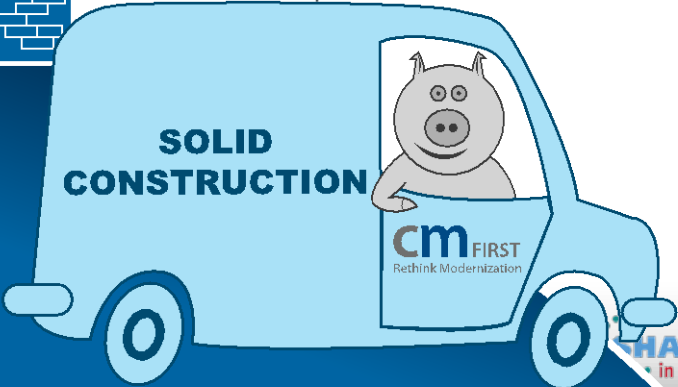

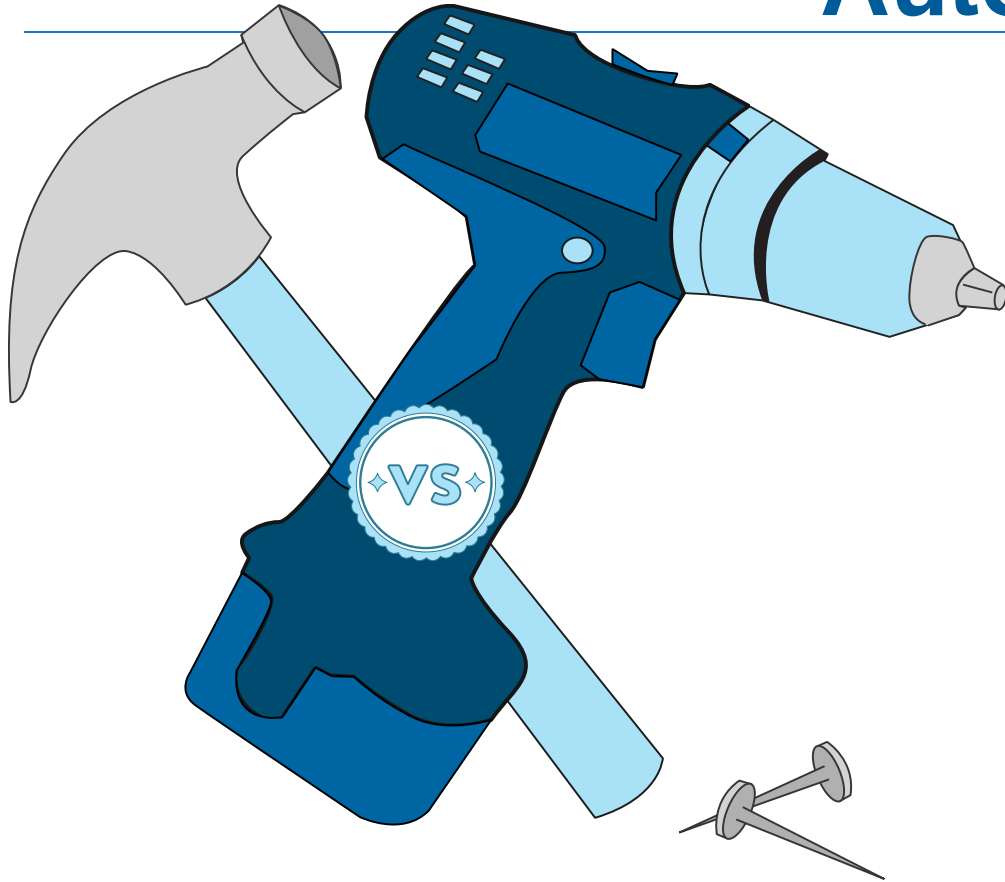Core Code Trend Analysis

Custom Code Trend

Solving The Problem - Smarter
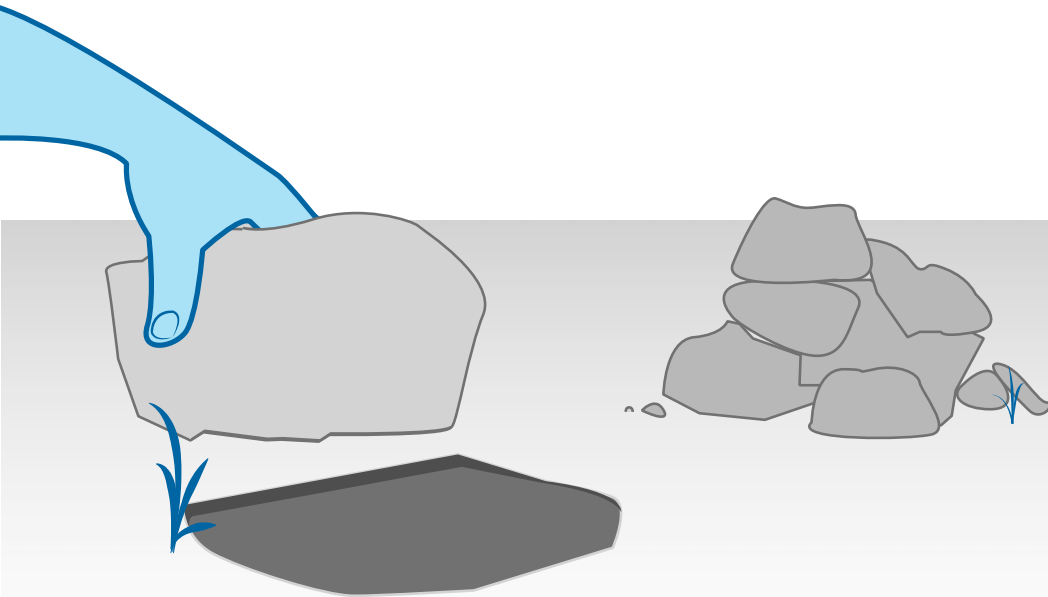
What do you need to look for?
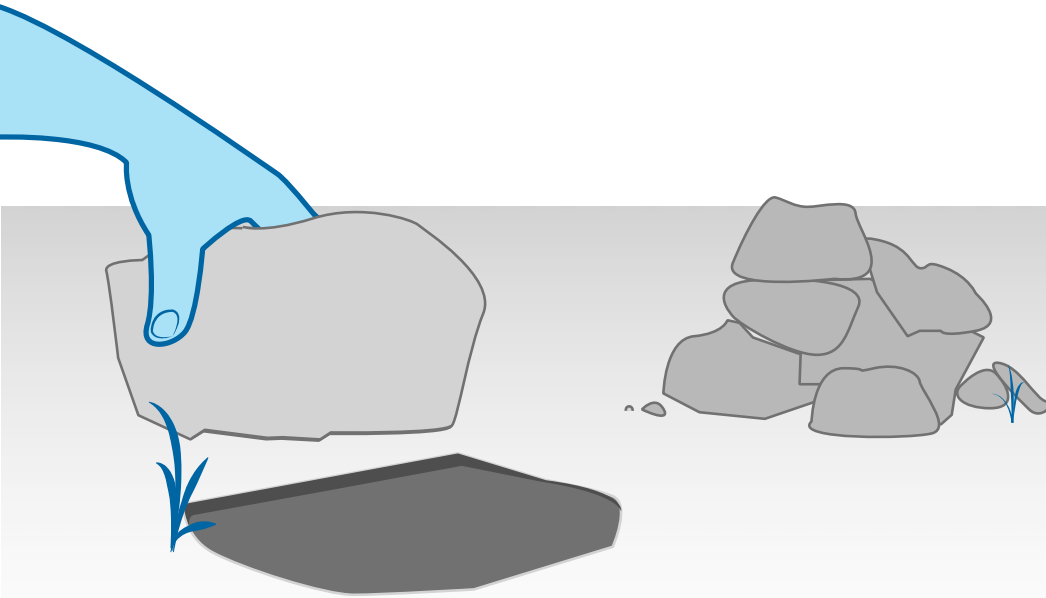
Automation Required

- Impossible for humans to manually analyze large software systems
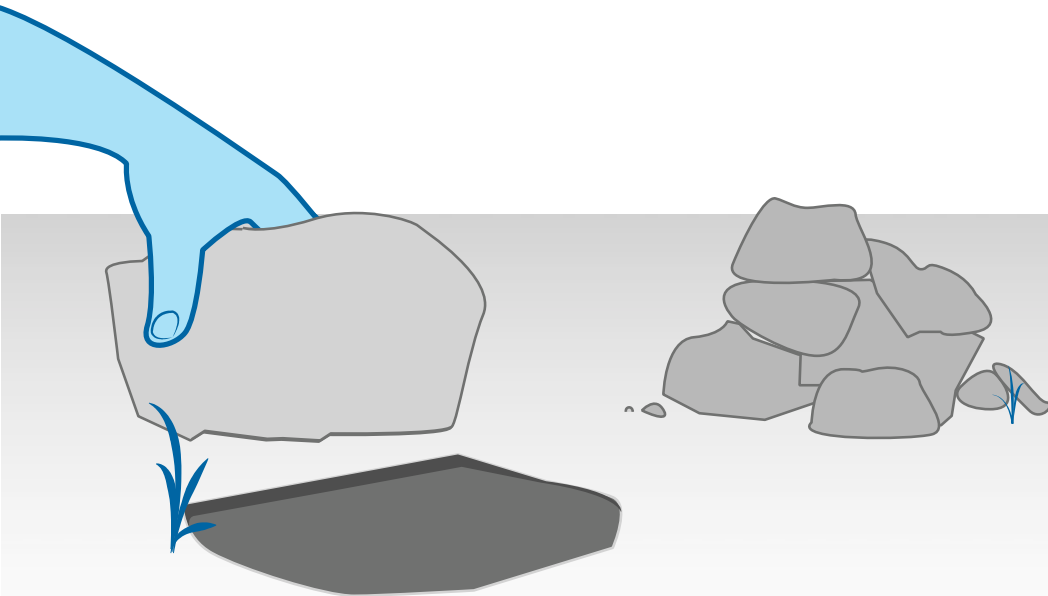
Does the solution cover the basics?

- Component Connectivity / Where Used
- Metrics
- Clones
- Test Coverage
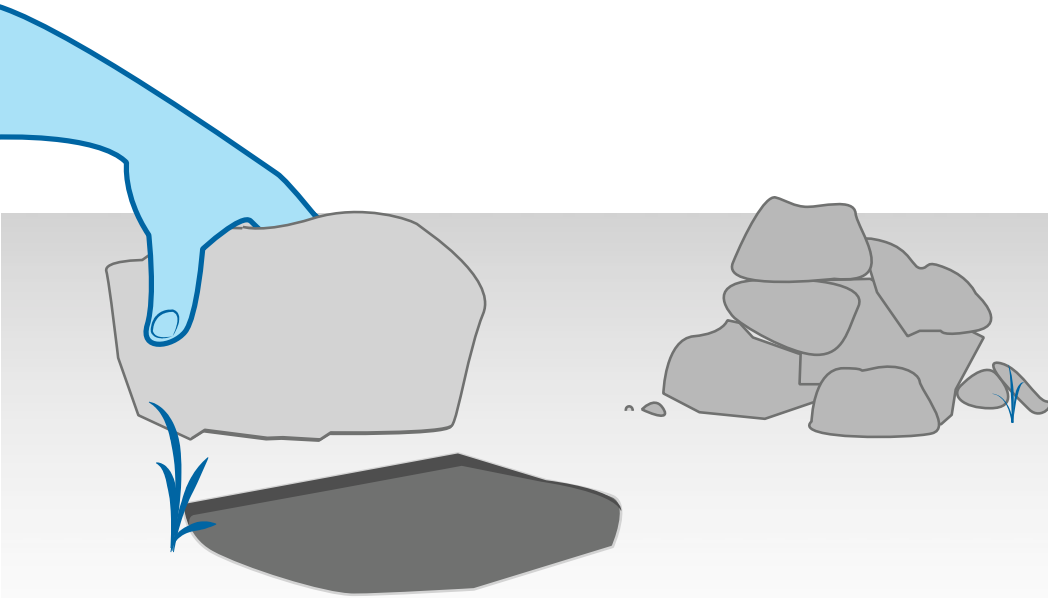
Is the solution enterprise grade?

- Link to SCM's like Endevor, 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

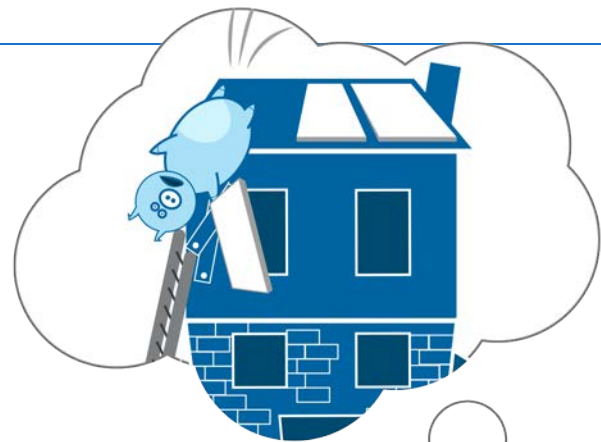# Does the solution cover your code base?

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

Questions / Feedback

Summary

- **Technical Debt can be costing your company**
  - **Outdated Documentation**
  - **Older Architectures**

- **Solving the Problem**
  - **Metadata repositories**
  - **Automated Discover**
  - **Code Visualization**
  - **Quality Metrics**
  - **Test Coverage**
  - **Clone Remediation**

SHARE
in Boston

SHARE

#SHAREorg

cm FIRST
Rethink Modernization

So next week…

Can you afford to do nothing?

john.rhodes@cmfirstgroup.com

Booth 231

SHARE in Boston   SHARE   #SHAREorg   cm FIRST Rethink Modernization