Technical Debt
The Cost in Performance & Security

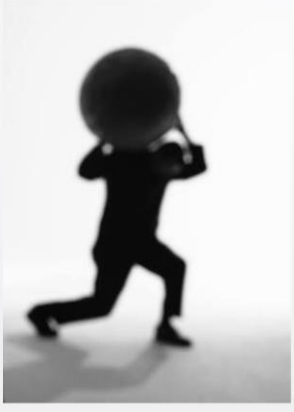
CM FIRST
Rethink Modernization

John Rhodes, CTO, CM First

Denise P. Kalm, Chief Innovator,
Kalm Kreative, Inc.

Agenda

- What is technical debt?
- How did we get there?
- What is it costing us?
- What can we do about it?



cm FIRST
Rethink Modernization

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

cm FIRST
Rethink Modernization

3

I am CTO of CM First Group, we are focused on helping folks modernize IBM System i and System z applications. I started my career as a COBOL and ASM programmer for Kraft

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

cm FIRST
Rethink Modernization

4

I am a long-time performance and capacity techie who is now helping companies spread their message thru quality writing and messaging.

What is Technical Debt?

cm FIRST
Rethink Modernization

"The cost of technical debt is the time and money it will take to rewrite the poor code after you ship and bring it back to the quality required to maintain the software over the long haul."

Chris Wysopal,CTO of Veracode

cm FIRST
Rethink Modernization

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

cm FIRST
Rethink Modernization

7

You may have started with a sturdy foundation, but if you don't keep building the same way, you don't get a good result. These are some of the general things that spawn debt.

The Metaphor

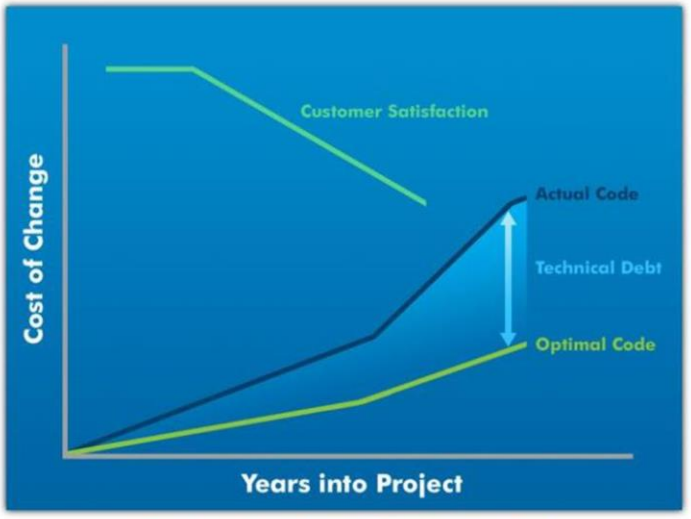
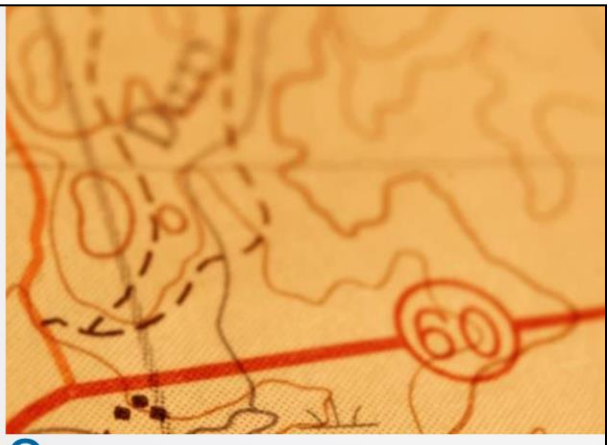


Fig. 1 - The Cost of Technical Debt

cm FIRST
Rethink Modernization

Actually Technical Debt was introduced as a metaphor by Ward Cunningham to help in these kinds of discussions. The idea is that code with a bad quality is like a financial burden. The total amount of the debt is the effort it would take to clean up the code base. The interest rate is the reduced productivity due to the bad code quality. Management is usually familiar with terms of the financial domain - so it should be easy to talk about software quality using this metaphor.

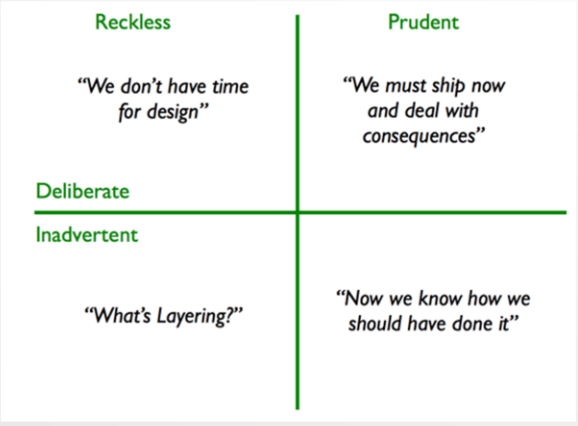


How Did We Get Here?

cm FIRST
Rethink Modernization

What journey were we on that got us to this point?

Technical Debt Quadrant



CM FIRST
Rethink Modernization

Martin Fowler,
http://martinfowler.com/bliki/TechnicalDebtQuadrant.html

10

Martin Fowler groups types of debt here – in each case, the cost can be high to the company, but the cause is different

Del-Reck – Crazy deadlines, too much pressure – worst kind

Del-Prudent – Very common – used when competitive pressures exert themselves – ship it now. Still costly. These two are made by startups all the time. It's part of the 'lean' mindset.

Inad-Reckless – Where you don't know enough to do it right – common with outsourcing

Inad-Prudent – Monday morning quarterbacking – when you figure out what it should have been.

General

What is it Costing Us?

cm FIRST
Rethink Modernization

Analysis is one category. Everyone likes to document and diagram, right? The fact is if you don't have documentation, it is extremely difficult (if not impossible) to find the 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

cm FIRST
Rethink Modernization

13

60% enhancements – 17% bug fixes of costs

Opportunity Costs

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

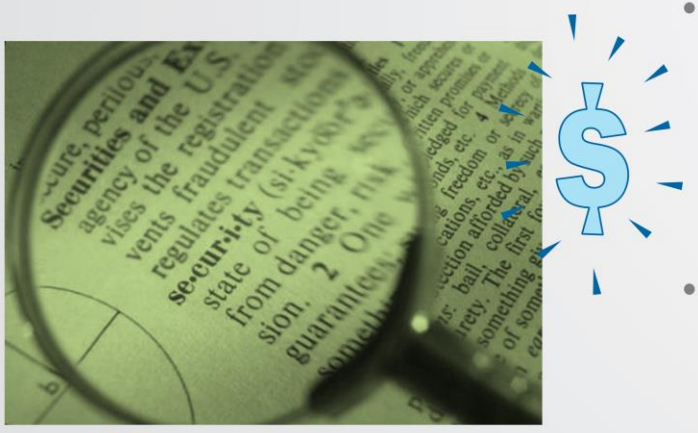
FIRST
Rethink Modernization

14

This is the biggest cost – but hard to quantify. If it takes an extra 6 months to develop a system to meet a new business opportunity, if that means a new insurance product

Companies like Caremark that have stated they will be increasing dividends to shareholders as a result of the affordable care act.

Compliance Costs



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

cm FIRST
Rethink Modernization

15

In the case of oil and gas companies – fine and imprisonment for 5 years for violations.

Do the math

"Intel reports saving $6 for every one $1 spent on meta data management."

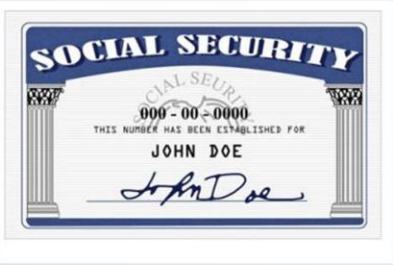
Source, ComputerWorld

cm FIRST
Rethink Modernization

16

Everyone must do the cost calculation for themselves, but often technical debt is something that can return money back to the company. For example, Intel had made some public announcements that they had saved

Social Security Administration & FSTAP



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

cm FIRST
Rethink Modernization

17

SSA convened the Future System Technology Advisor panel (FSTAP) in 2010 to look at modernization challenges. Needed to migrate their customer database system to DB2. But this spawned the discussion of what else they needed to modernize. They had some challenges. Their code base was 30+ years old, they had 250 MM lines of code running on over 250 major systems, 17 of which were mainframes, and 93% of their budget was dedicated to maintenance. And time was not on their side. They had to make major changes to Medicare and Medicaid.

Security

What is it Costing Us?

cm FIRST
Rethink Modernization

Trustworthy Computing is computing that is as available, reliable and secure as electricity, water services and telephony.
Bill Gates, Microsoft

cm FIRST
Rethink Modernization

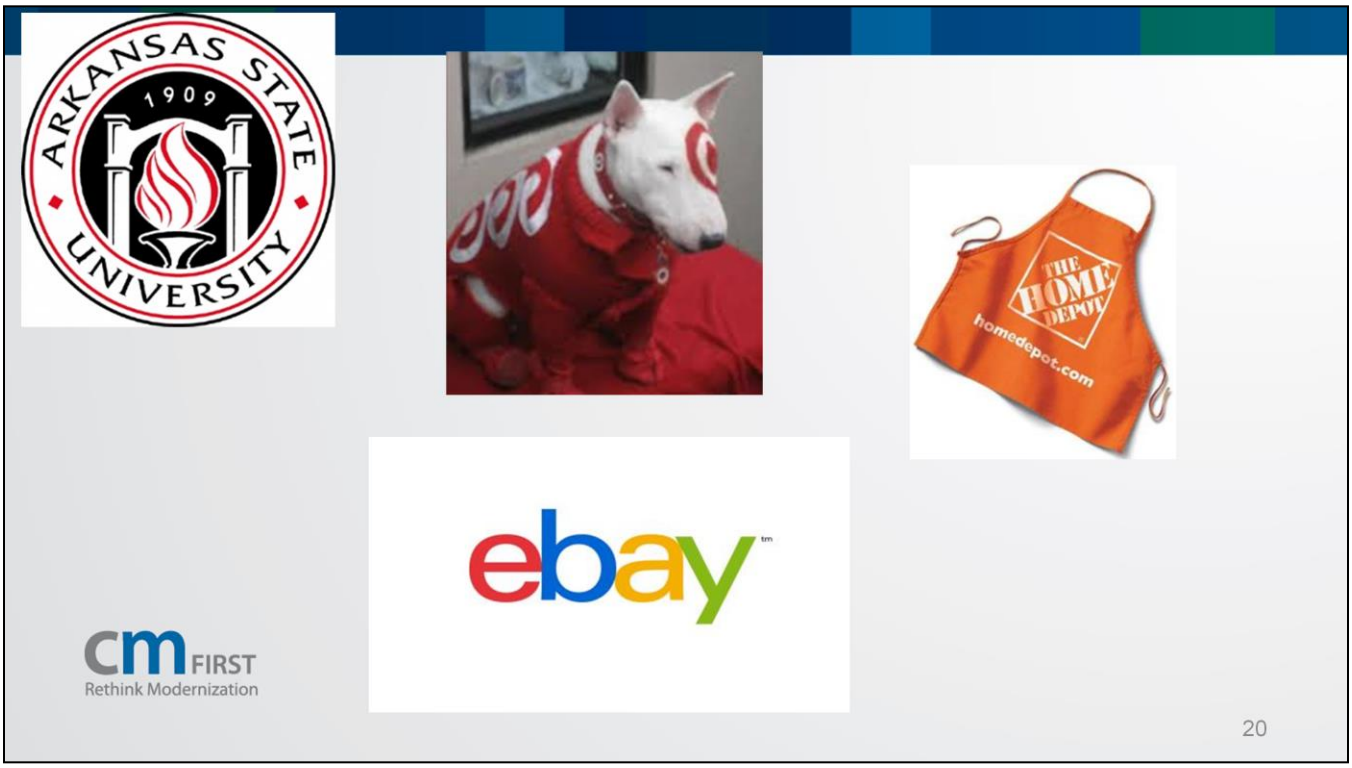
19

People expect security in computing. In his 2002 Trustworthy Computing memo to employees and subsidiaries,Gates focused on 3 areas:

Availability: r products should always be available when our customers need them. System outages should become a thing of the past because of a software architecture that supports redundancy and automatic recovery. Self-management should allow for service resumption without user intervention in almost every case.

Security: The data our software and services store on behalf of our customers should be protected from harm and used or modified only in appropriate ways. Security models should be easy for developers to understand and build into their applications.

Privacy: Users should be in control of how their data is used. Policies for information use should be clear to the user. Users should be in control of when and if they receive information to make best use of their time. It should be easy for users to specify appropriate use of their information including controlling the use of email they send.

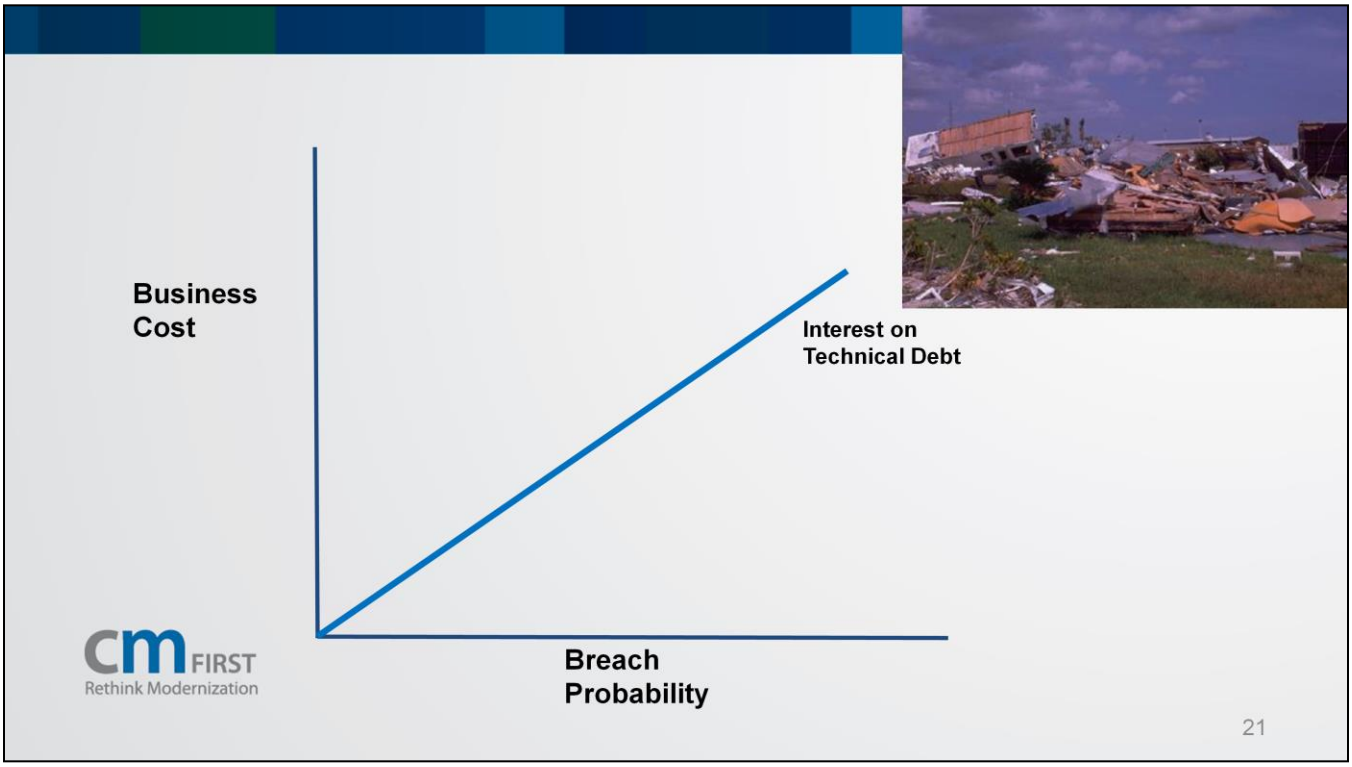


ASU – 50K Social Security #'s stolen

Target – 40 MM payment cards – info released

Home Depot – 30K accounts compromised

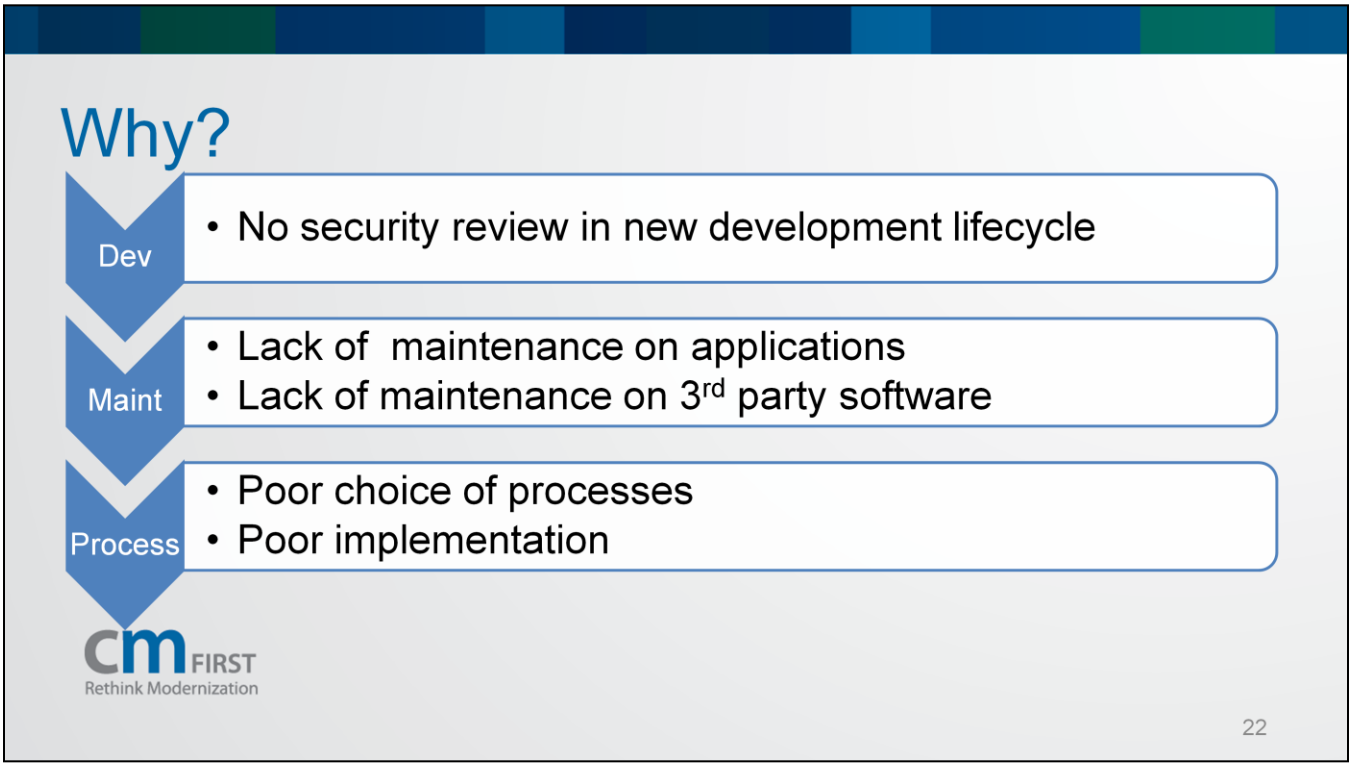
eBay – 145MM customers compromised



You don't pay interest on security debt if no one saw the exposure before you fixed it. But the cost goes up with the probability of a breach. The challenge with this kind of technical debt is it requires more than simply recoding. You often must change processes and may also have to make administrative changes. Costs are:

Detection & Escalation, Notification, Ex-Post Response, Lost business (Chris Wysopal at Veracode)

"Application Security Debt is a 'loan' with variable principal which could range from 0% to 100% of your original project costs. The 'principal' is what you'll eventually have to pay to fix security bugs or rewrite the code. It also has varying and uncertain 'interest costs', which are the costs of security breaches due to these vulnerabilities. This includes the possibility of the mother-of-all balloon payments (i.e. a huge loss event)."



Security debt comes from some of the same places as regular tech debt, but it also has these issues. Which means these are the 3 areas you need to consider in remediatiaon.

"It has been like trying to repair the engine on a running car on a racetrack – impossibly difficult and ultimately futile."
Oliver Rochford, author

CM FIRST
Rethink Modernization

23

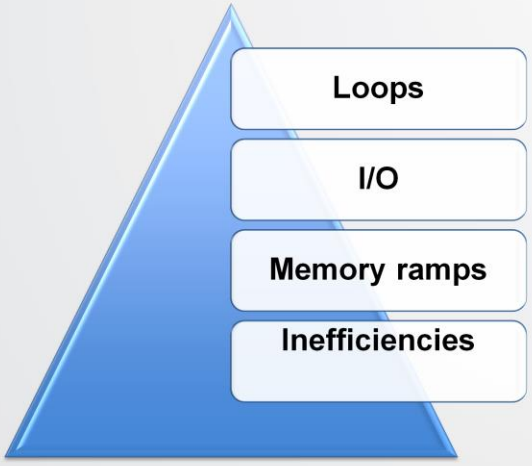
Rochford notes that the during the tech and debt bubble, people went nuts rushing things to market. The complexity and scale were unprecedented. Then, when the bubble burst, companies cut so deeply they didn't have the money or people to fix anything. It's hard to get budget holders to invest money in something that doesn't bring in money or customers. Before, we didn't have time to do it right; now we don't have the money.

Performance

What is it Costing Us?

cm FIRST
Rethink Modernization

Some Types of Performance Debt



cm FIRST
Rethink Modernization

25

Looping code is a bug, but if it is an area that isn't hit all the time, it may not show up, or it may loop only under certain circumstances

I/O is always slower than CPU – so you want to do as little as possible. Examples of poor coding include reading in the same record multiple times, using inefficient key mechanisms, poor RDMS design, index problems and more. One application I know did more than 200 I/Os per txn – and there were only 16 files!

Memory ramps – misuse of memory affects all txns in the system and can be hard to detect. Memory pigs cost you more as the business peaks

Coding inefficiencies include poor design of if-then-else, voodoo code (maintained, but not useful functions), calling functions more than once, fixing bugs by writing code that overwrites the result of faulty code, and other problems like JITed code

Poor design from the beginning is another area. Can't design OLTP if it comes out like batch.

Performance debt comes about both because of lack of knowledge on how to write efficient code and also from taking shortcuts that are 'long-cuts' when it comes to execution.

Causes

- Can’t map from A-Z
- Not comfortable with programming language
- Not comfortable with platform
- Inability to comprehend pointers
- Difficulty seeing through recursion
- Distrusting code
- Copying bad code

cm FIRST
Rethink Modernization

26

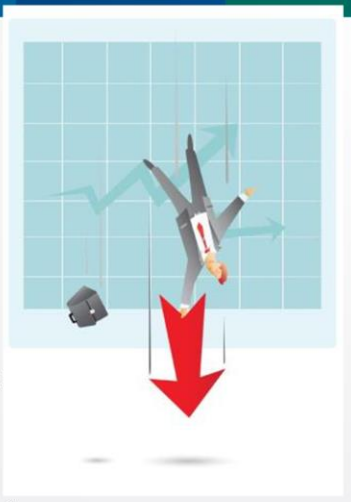
Poor ability to map a clear/short path from point A to Z

Poor understanding of the language's programming model

Deficient research skills / Chronically poor knowledge of the platform's features

The concept of pointers enables the creation of complex data structures and efficient APIs. Managed languages use references instead of pointers, which are similar but add automatic dereferencing and prohibit pointer arithmetic to eliminate certain classes of bugs. They are still similar enough, however, that a failure to grasp the concept will be reflected in poor data-structure design and bugs that trace back to the difference between pass-by-value and pass-by-reference in method calls.

Checking and checking again on something



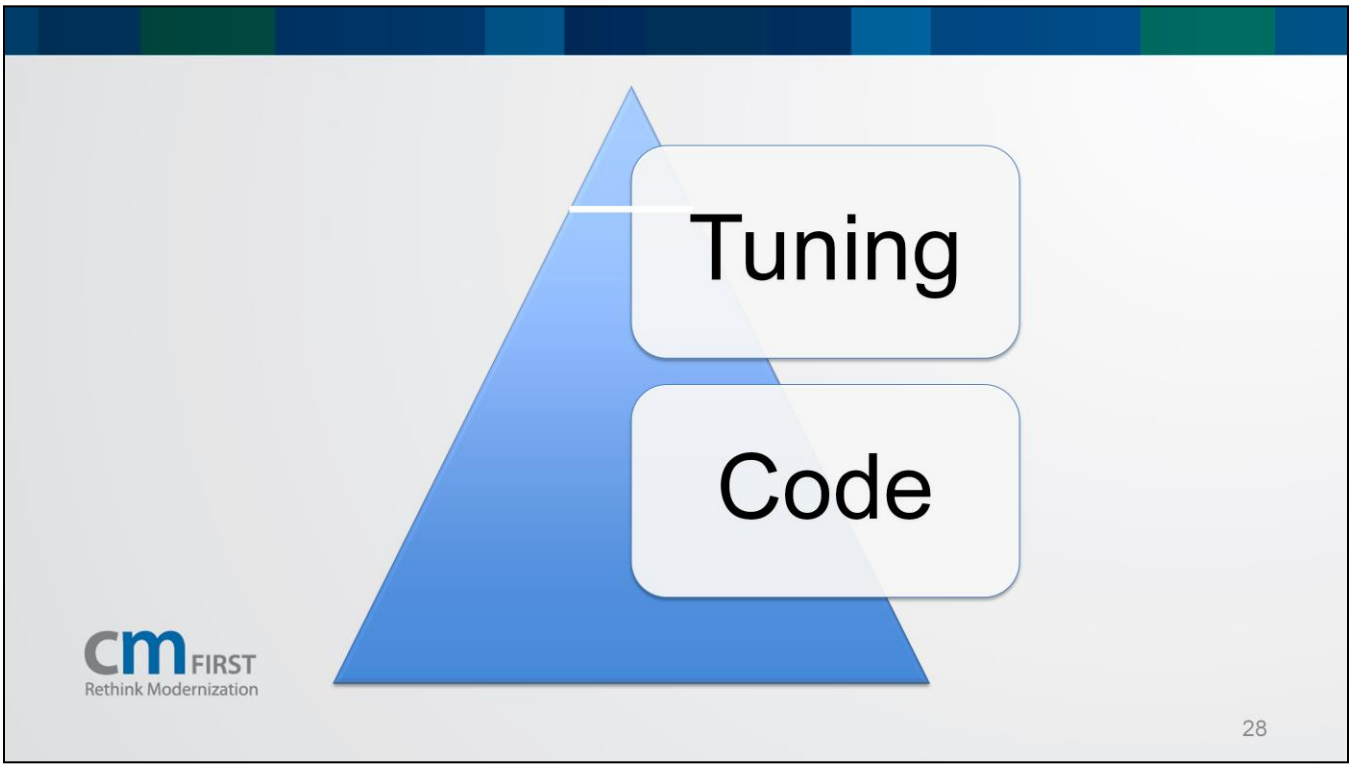
"The biggest factor affecting application performance today is poorly optimized code and infrastructure, such as suboptimal SQL queries, poorly configured network infrastructure, or inefficient code algorithms at the application layer."

Rick Houlihan,
VP of Engineering, Boundary

CM FIRST
Rethink Modernization

27

You can't afford bad performance or worse, availability issues. Customers aren't loyal anymore. Provide them with sub-optimal service and they will find someone else to buy from. Your sales will plummet and you can very well go out of business. It's that important. And yet, the seeds of your destruction likely lie mostly in the code you take for granted. The code written a while back, in a hurry, or by people who didn't know what they were doing.



Impact of tuning is much less than the impact of fixing code. At least 80% of performance is baked in by inefficient or simply bad coding.

What Can We Do About It?

cm FIRST
Rethink Modernization

Rescue is at Hand

15 months

- Evaluate
- Analyze
- Remediate

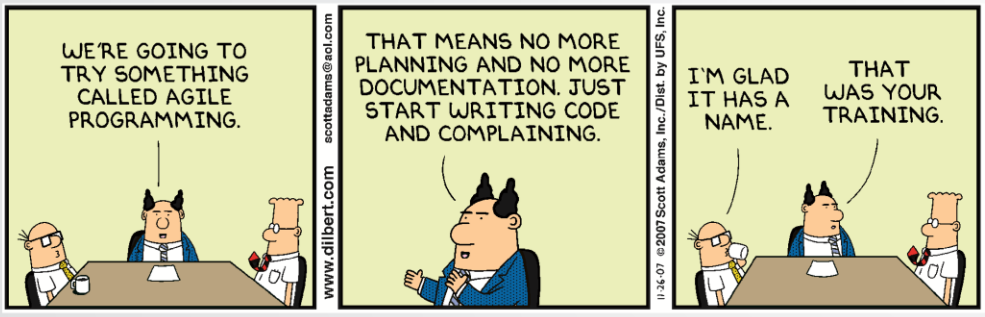
100% Automated

CM FIRST
Rethink Modernization

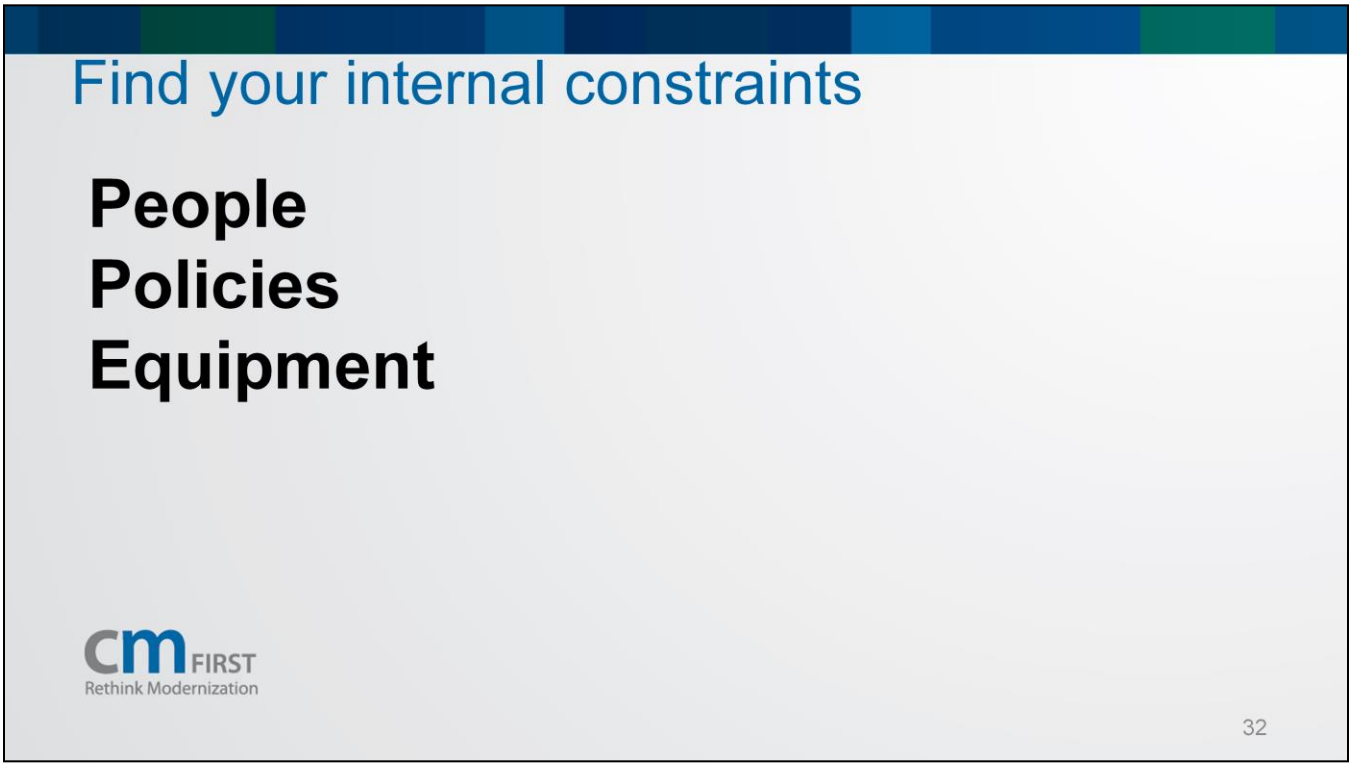
30

SSA saw a solution they liked at CA World 2009. In only 15 months, they were able to complete a POC which included req analysis, development of custom features by the vendor, installation, configuration and integration. Success meant they wanted to continue to work with the vendor on their 10 year modernization plan. Which is good news for all of us hoping to retire someday.

The easy way to add agility to business software!



CM FIRST
Rethink Modernization



There can be many constraint in business. External and External.

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

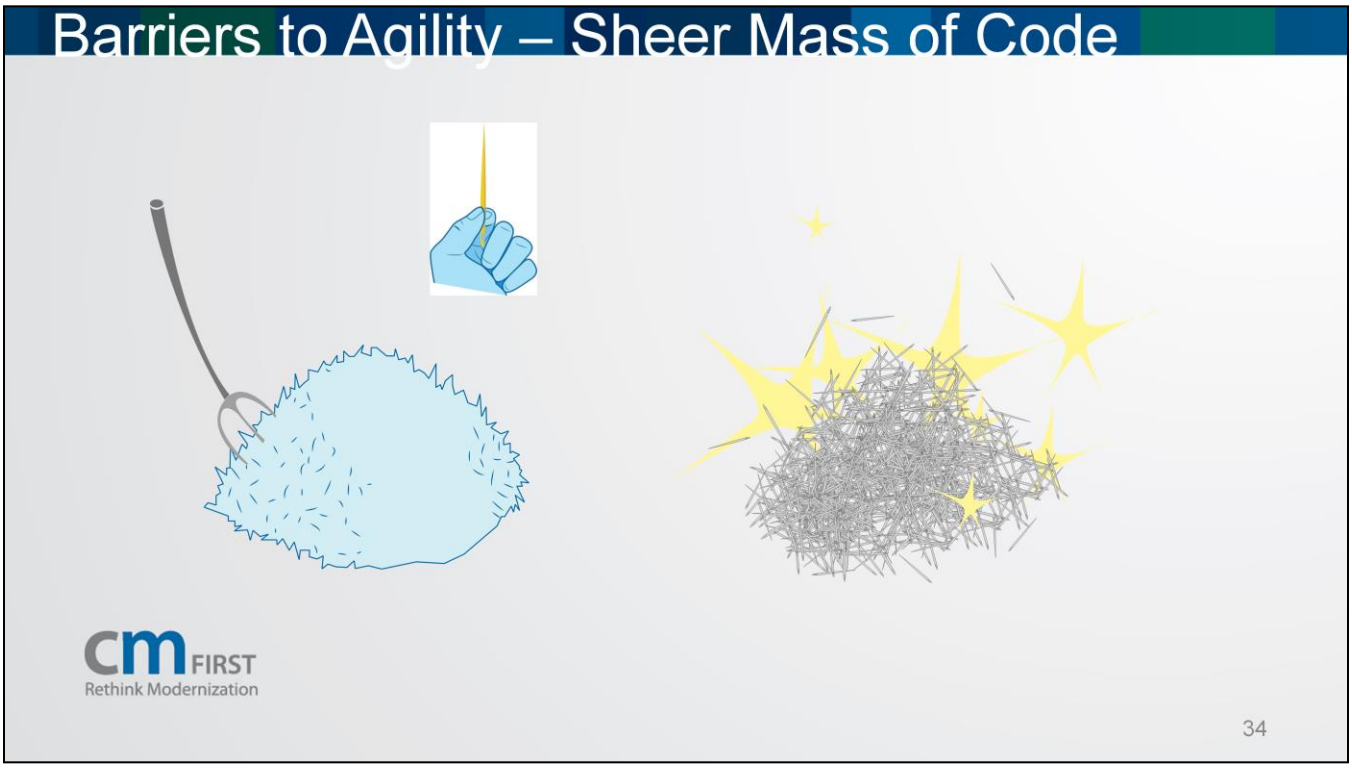
cm FIRST
Rethink Modernization

33

Most rewrites fail – capturing the business rules, and new requirements is very difficult. Package implementation

Adding more resources – the old

Like painting the bridge – using substandard paint



Why can't we move forward – adages we here –

it is like finding a needle in a haystack

Or like finding a needle in a stack of needles

Or like boiling the ocean

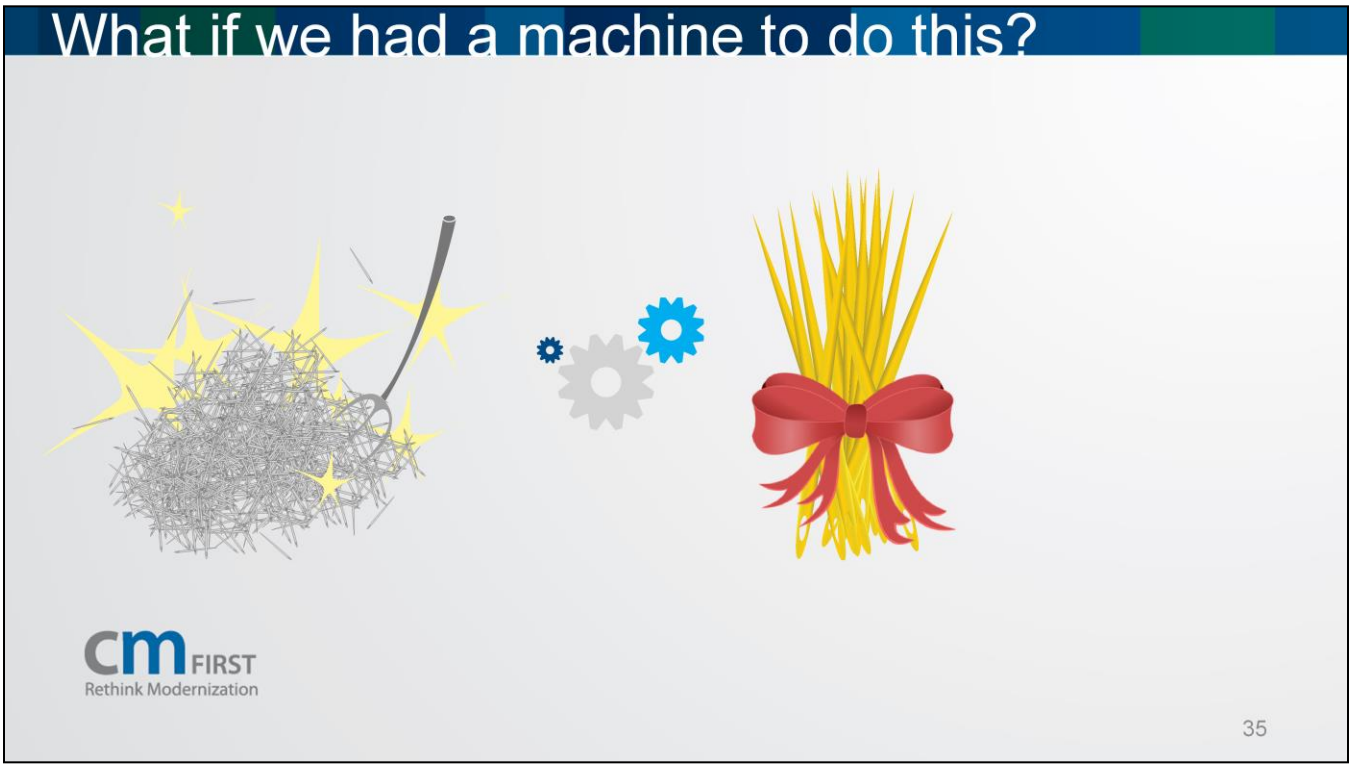
Some of the numbers we come across on from customer projects are staggering

45,000 compiliation units

200 applications of 100 Lines of code

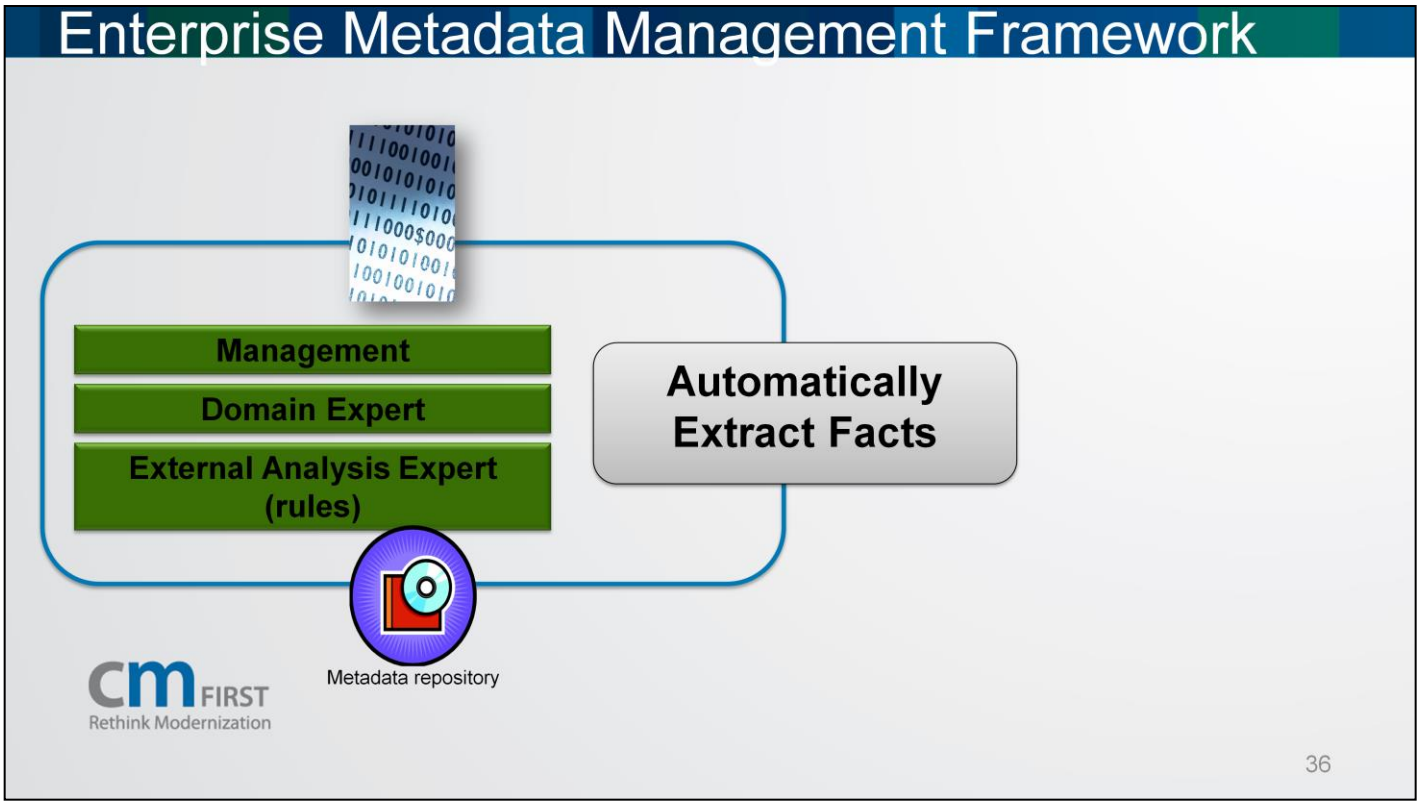
50 Million lines of COBOL

It takes about 12 24 hour days to just count to a million



You need to look at machinery to do the analysis for you

What if you had a machine to sort and package the facts you are looking for and package them neatly



I will submit this as a basic debt management framework that software development organizations should have in place.

Agile, iterative process

Automated Software Discovery

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

cm FIRST
Rethink Modernization

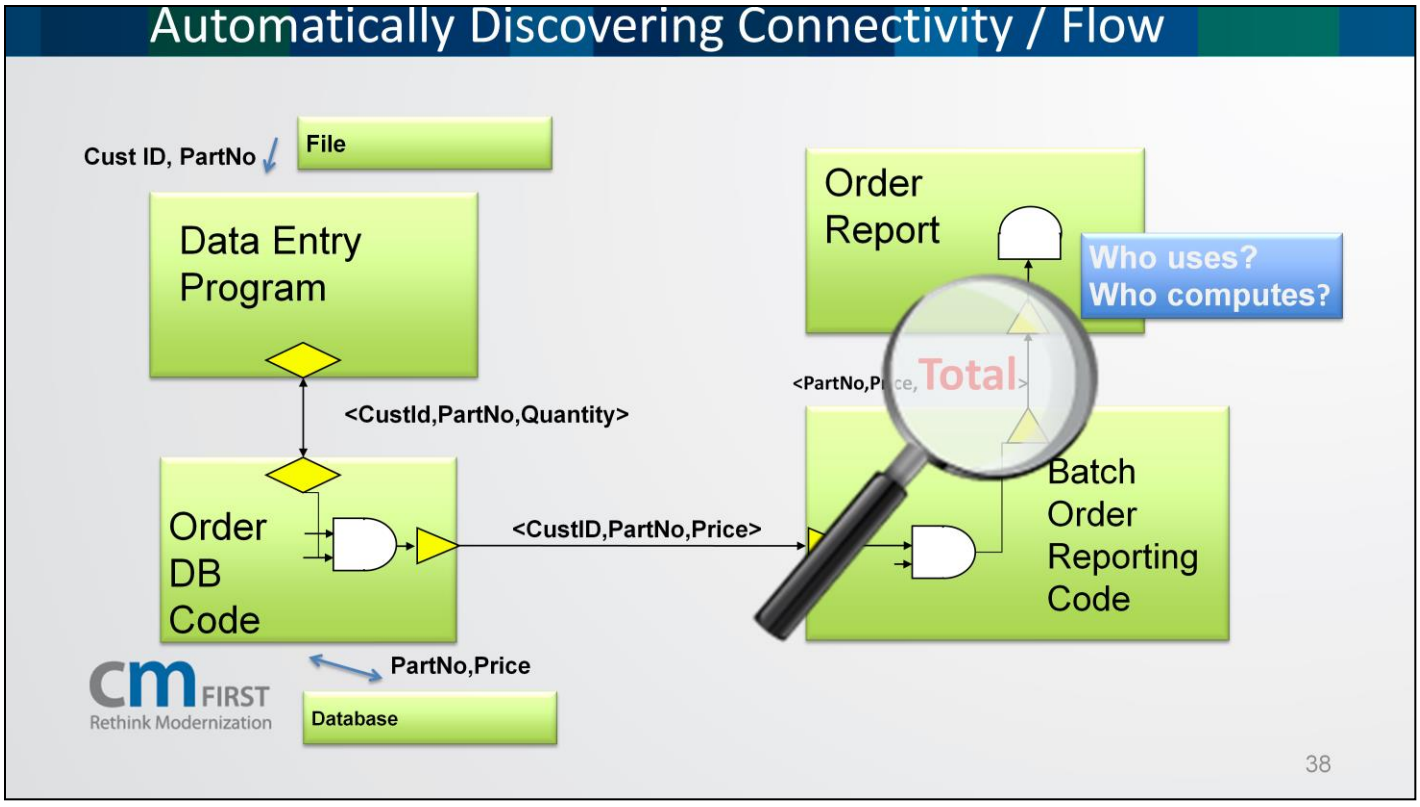
37

The discovery process works like this

Read your code bases – list off

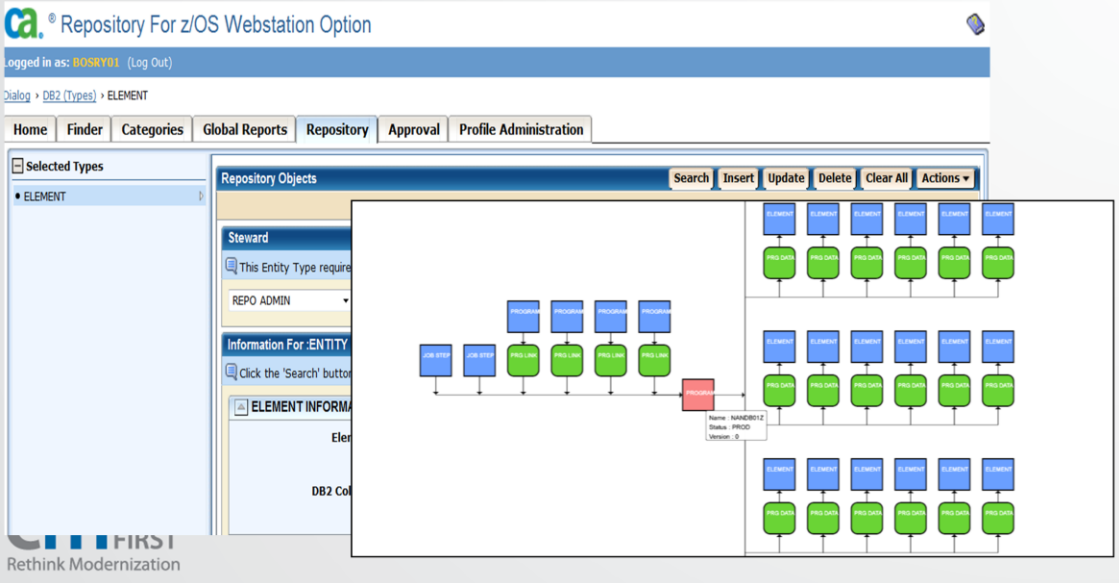
Extract facts

- Maintainabilty
- Connectivity
- Test Coverage
- Style checking



One important set of facts is software connectivity. Who talks to who? What files do they access

MetaData Repositories



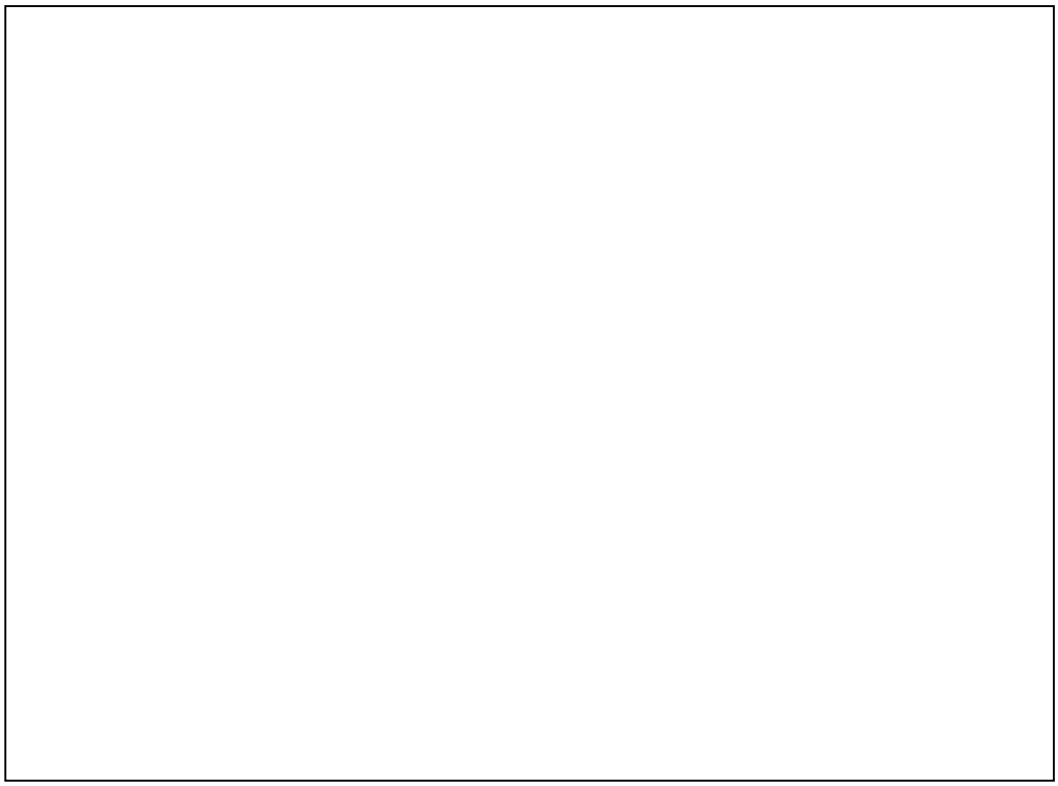
One of the repositories you can work with is CA Repository, which has a zOS based component. The main thing is to have a repository for the code

Code Quality Metrics

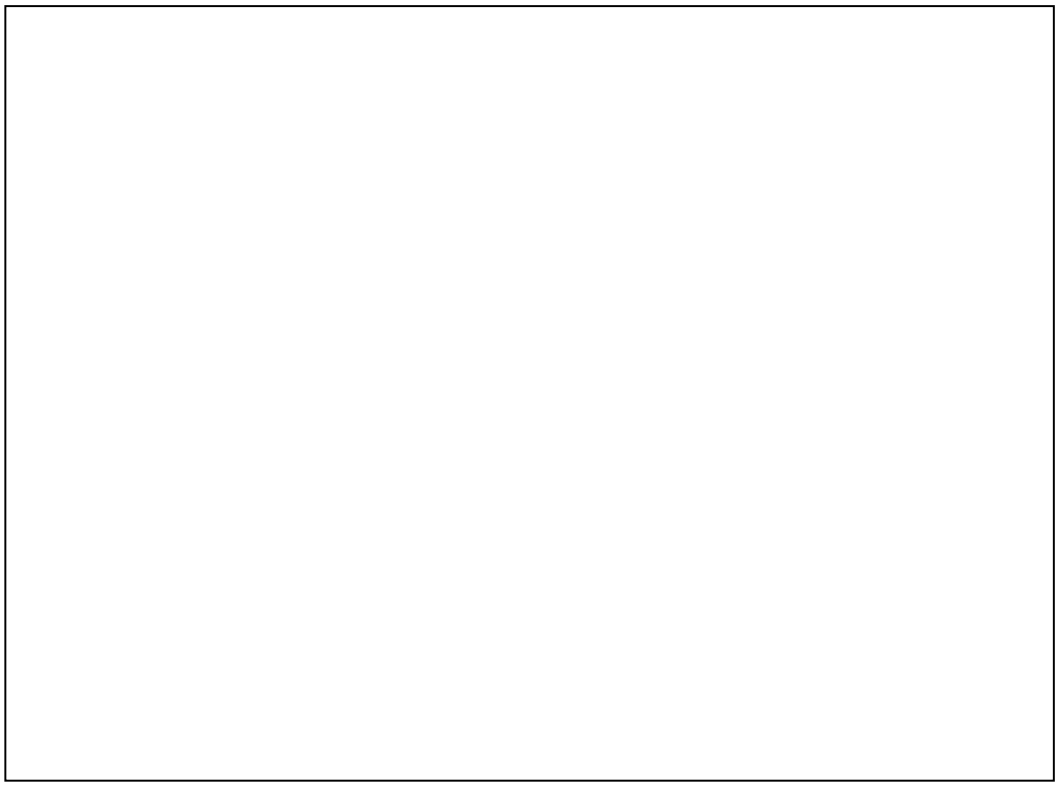
- Assess the quality of the code base
- Track and roll up
- Typical Metrics
  - Halstead/McCabe
  - SEI / Maintainability

cm FIRST
Rethink Modernization

Code quality is another key fact to understand. There are metrics to help understand this.



This is another metric showing complexity.



Example Analysis

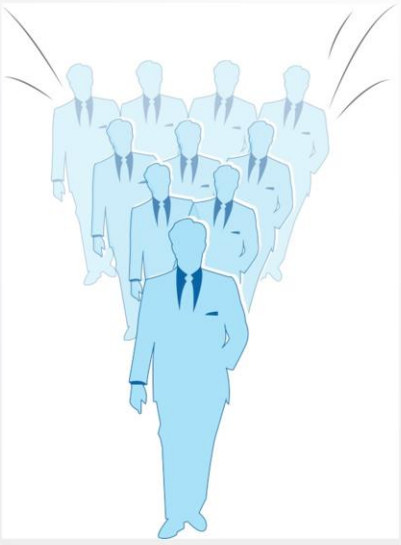
Source Lines	Code Lines	Comment Lines	Blank Lines	Cyclomatic Complexity	Halstead Complexity	Filename
1884	1884	0	0	497	4595350.5	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool Exports/ExportLargeProperty/Function/Function FirstAid Divisional
4618	4616	2	0	214	1.34E+07	C:/Document FirstAid Fire
2487	2479	7	2	204	4713399.5	C:/Documen FirstAid Invo
3161	3158	3	0	199	7425025	C:/Documen FirstAid Wor
1173	1172	1	0	184	1837039.2	C:/Documen FirstAid Divi
1430	1414	29	5	173	4855147.5	C:/Documen FirstAid New
2247	2241	6	0	170	3829034.5	C:/Document FirstAid Custo
2747	2732	14	1	166	8507429	C:/Documents and Settings/C1162332/My Documents/Plex/StellaTool 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

Rethink Modernization

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

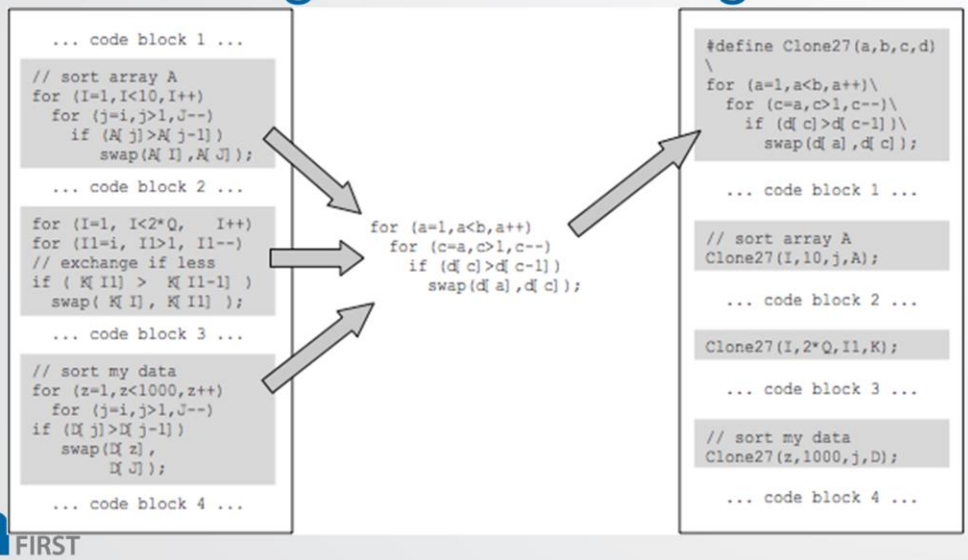


cm FIRST
Rethink Modernization

44

But this is the most common thing people do – copy code, making small changes but basically using old code. If it is good code, that's fine, but when the code is sub-optimal, this magnifies the problem.

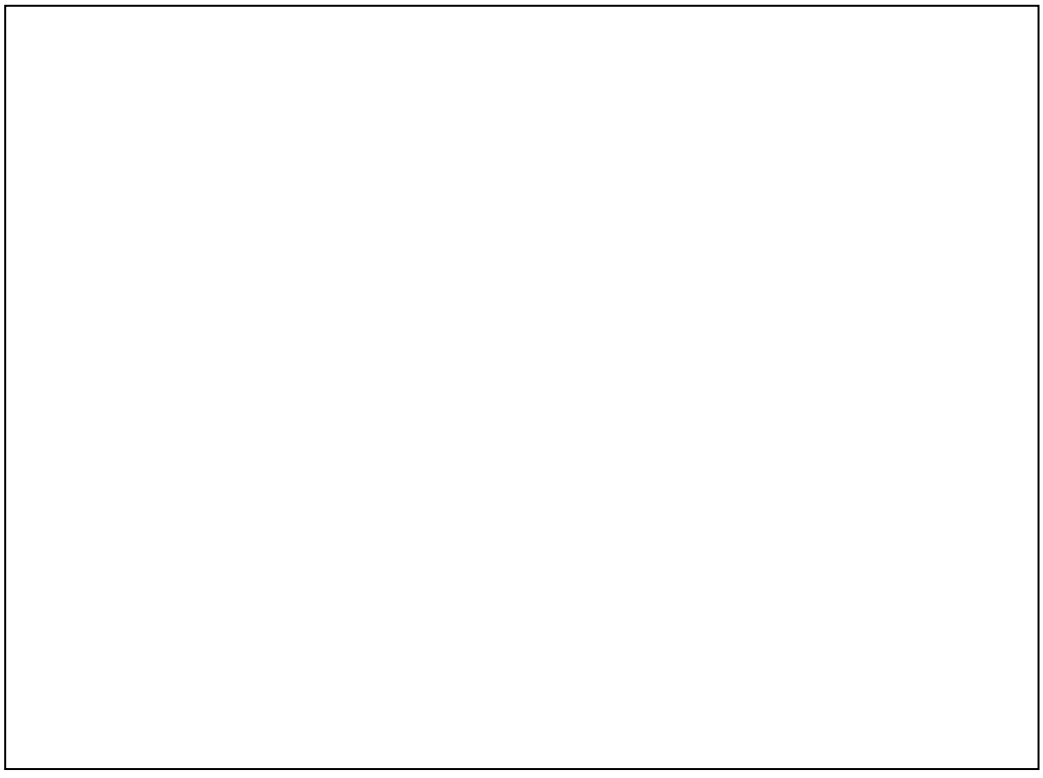
Detecting / Remediating Clones

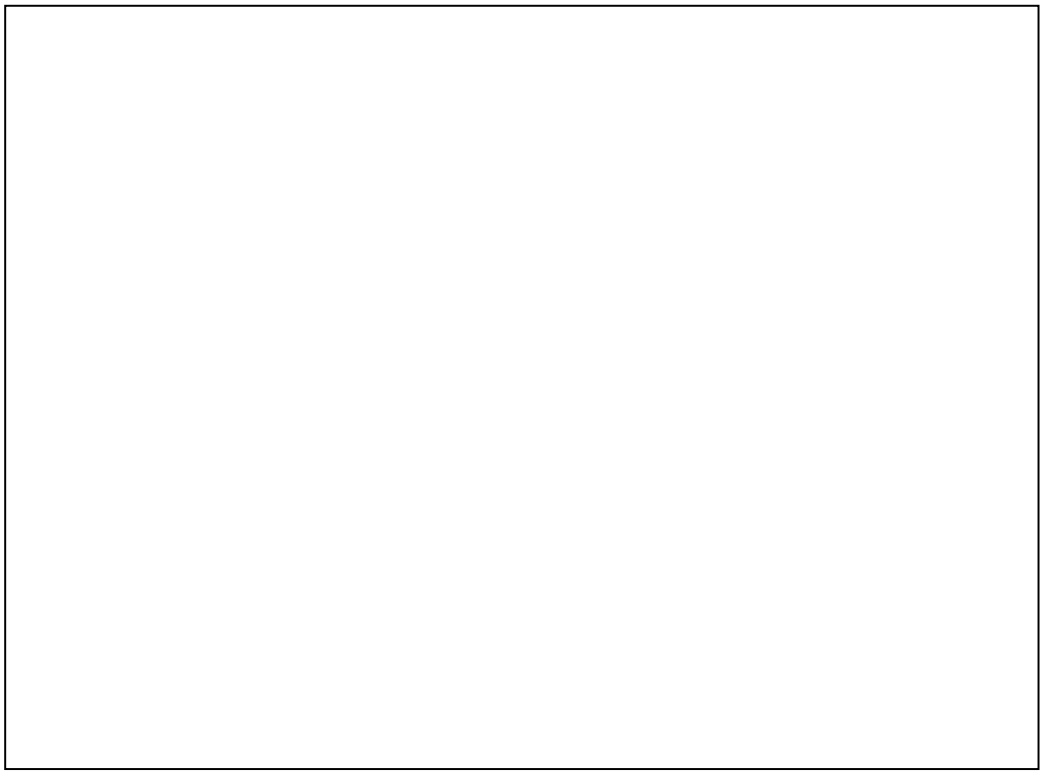


cm FIRST
Rethink Modernization

45

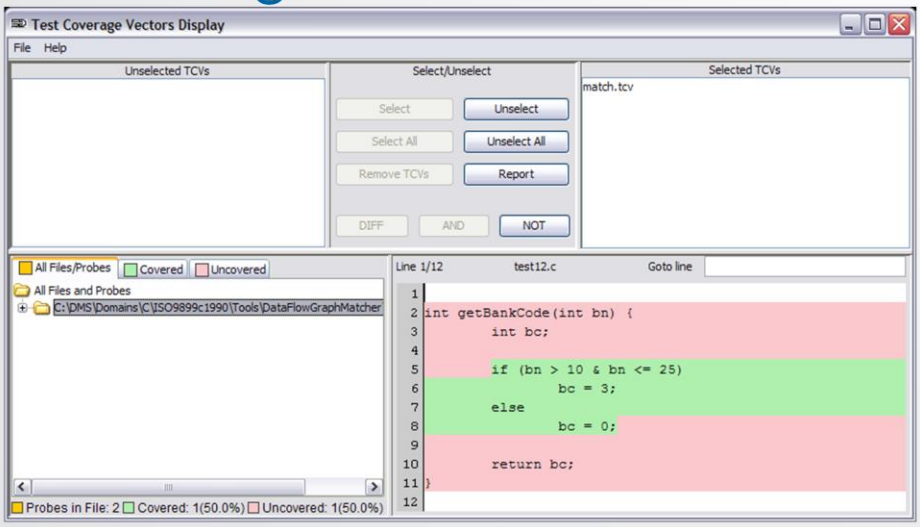
This shows how it can work.





Need an explanation here!!!

Data Flow Range Match



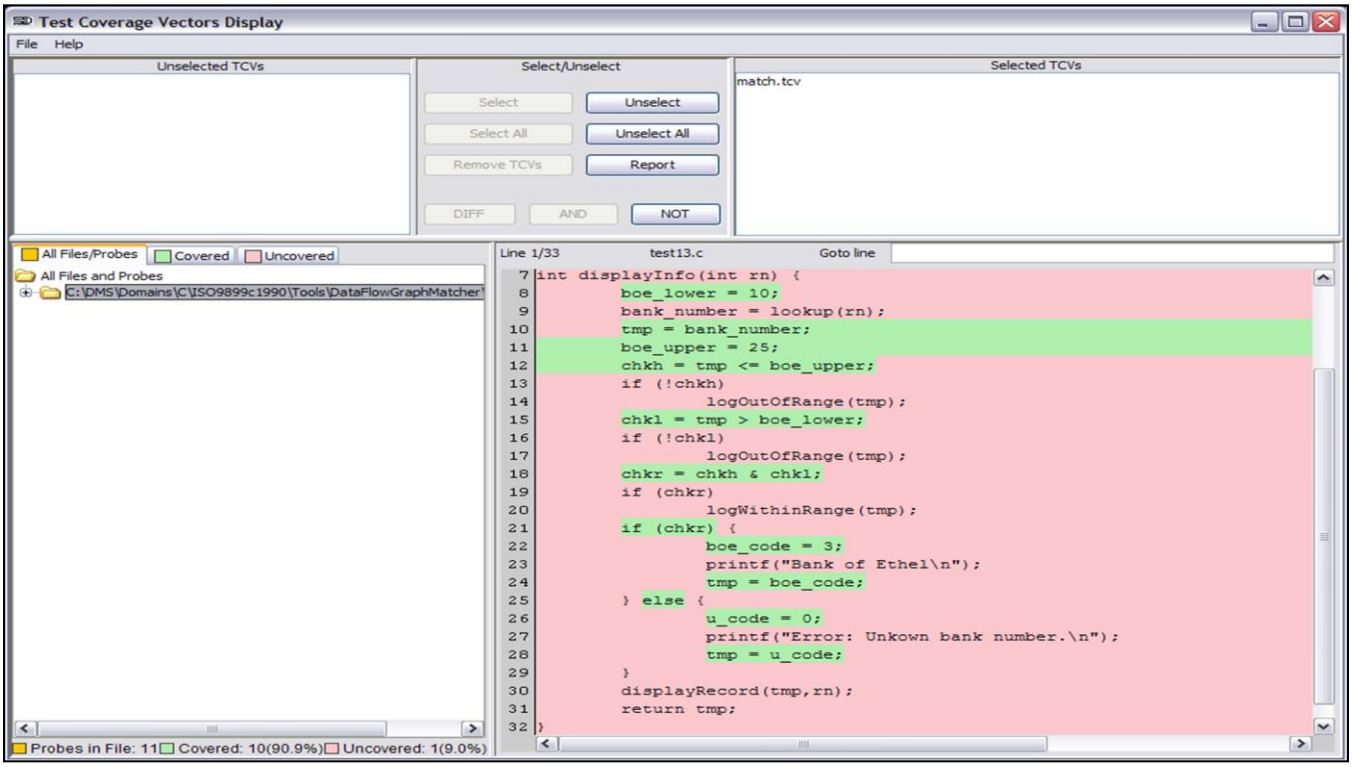
cm FIRST
Rethink Modernization

48

One challenge is to match arbitrary patterns using data flow. The first is a simpler program.

One (“simple”) proram obviously contains such a check (findable also by the SCSE tool) and the other (“complex”) which contains the check but it is scattered about in a bunch of small pieces, connected by dataflows (which is why this requires a dataflow matcher to do it well). In the complex case, the SCSE will not find the check because there is no obvious complete idiom; you can’t realistically hunt for all the “<=” in a code and assume they are bank related, without getting a huge number of false positives.

You can see the results of the match in the displays; the matching code is lit up. **The matching process and the display generation are completely automated.** We just fed the tool the data flow pattern and the code.



The more complex program.

Solving The Problem… Smarter

What do you need to look for?

Automation Required

What should your solution look like?

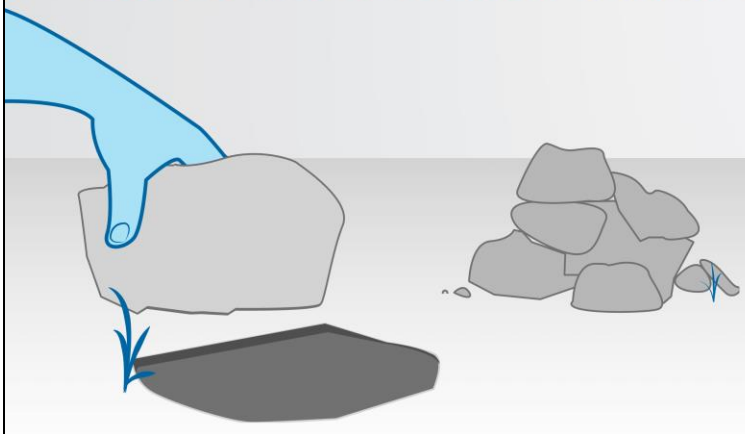
cm FIRST
Rethink Modernization

51

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?



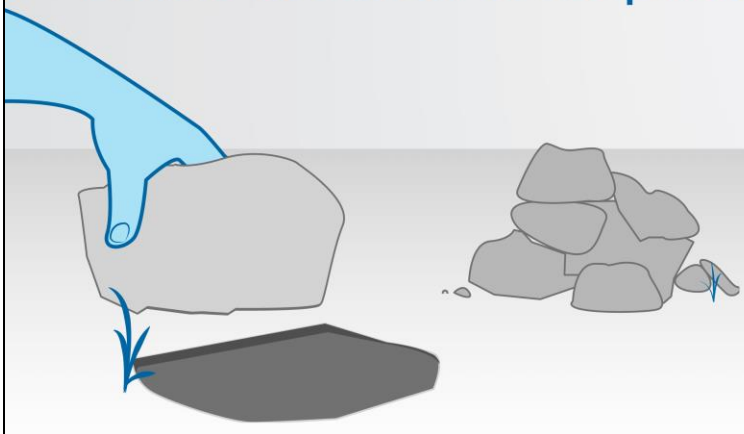
- Component Connectivity / Where Used
- Metrics
- Clones

Test Coverage

52

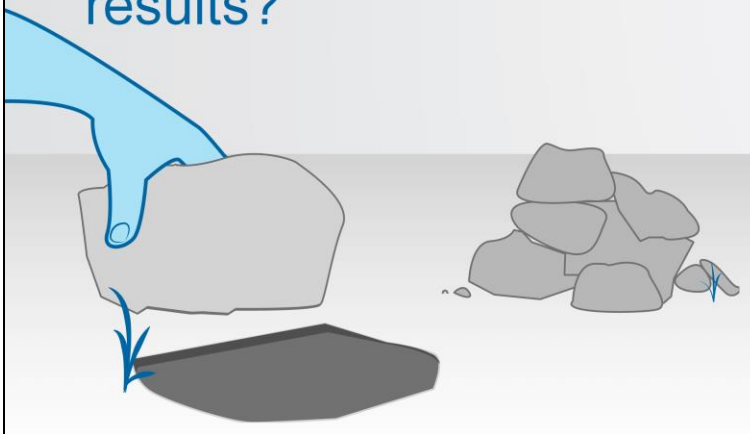
Questions to ask in your RFP. You need to know the following things to decide if the solution will work for you.

Is the solution enterprise grade?



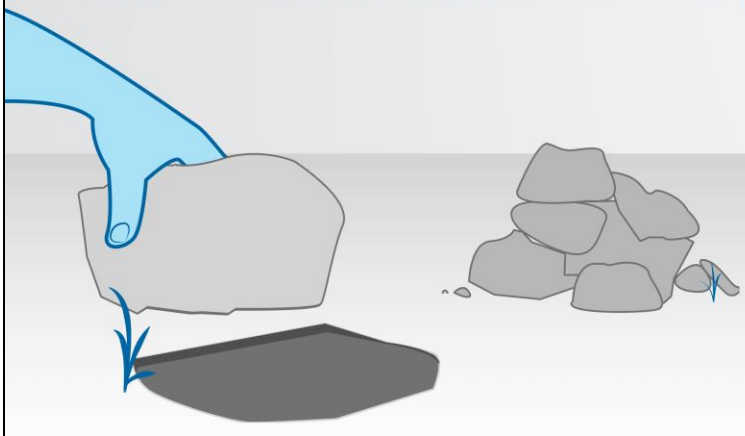
- Link to SCM's like Endevor, PTC
- Scan code base in daily window
- Handle millions of lines and resulting data in the EMR

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

D2

YouTube

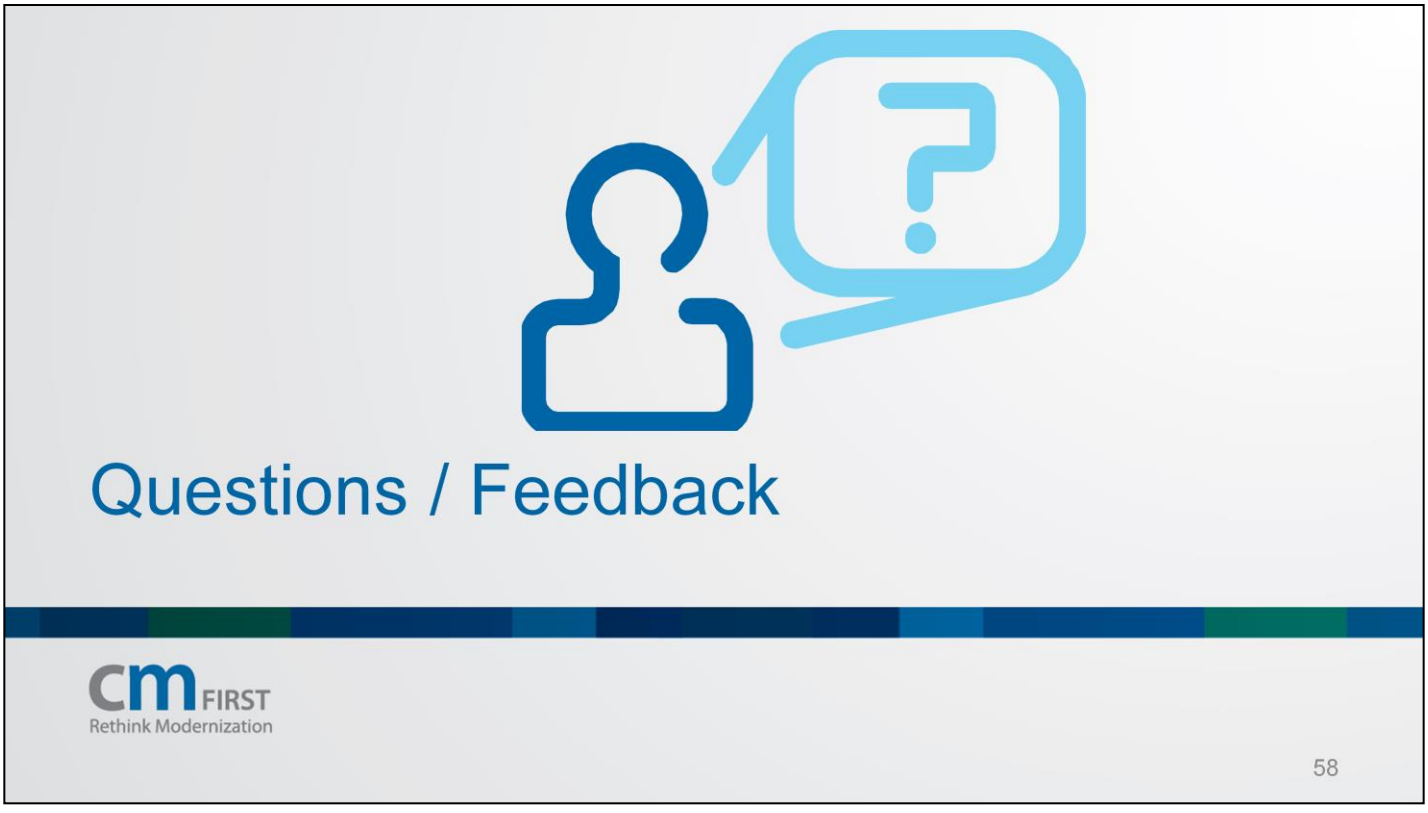
Summary Video

CM FIRST
Rethink Modernization

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

cm FIRST
Rethink Modernization



Before we conclude, are there any questions?

Isn’t it time to wake up and smell the technical debt in your code? Remediating technical debt while save on spending, improve performance, reduce security exposures, decrease risk and increase agility. Start next week!