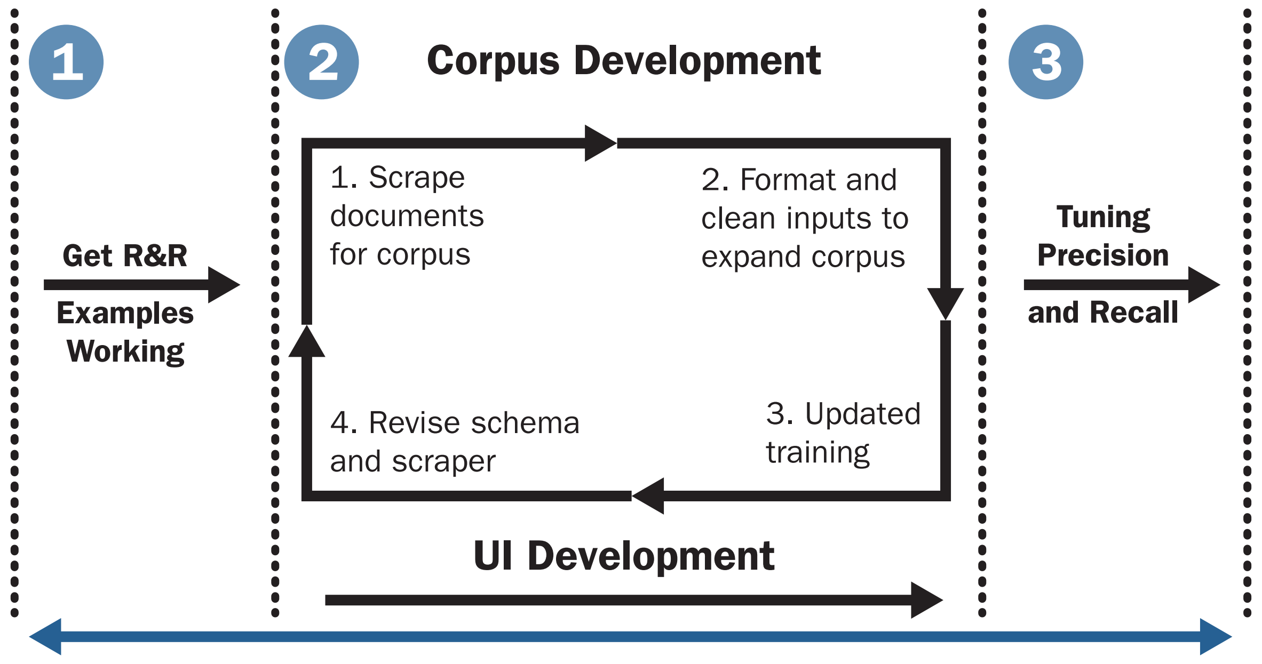
Developing and IBM Watson Cognitive Processing Application
Supporting Application Security (Software Assurance)

IBM Watson made an impressive introduction. In 2011, Watson competed on one of America's leading question and answer shows against former winners Brad Rutter and Ken Jennings. Watson received the first place prize of $1 million.*

Watson is a question answering computer system capable of answering questions posed in natural language, developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci. Watson was named after IBM's first CEO and industrialist Thomas J. Watson. The computer system was specifically developed to answer questions on one of America's leading question and answer shows.

Application development timeline


1
Get R&R Examples Working
2
Corpus Development
1. Scrape documents for corpus
2. Format and clean inputs to expand corpus
3. Updated training
4. Revise schema and scraper
UI Development
3
Tuning Precision and Recall

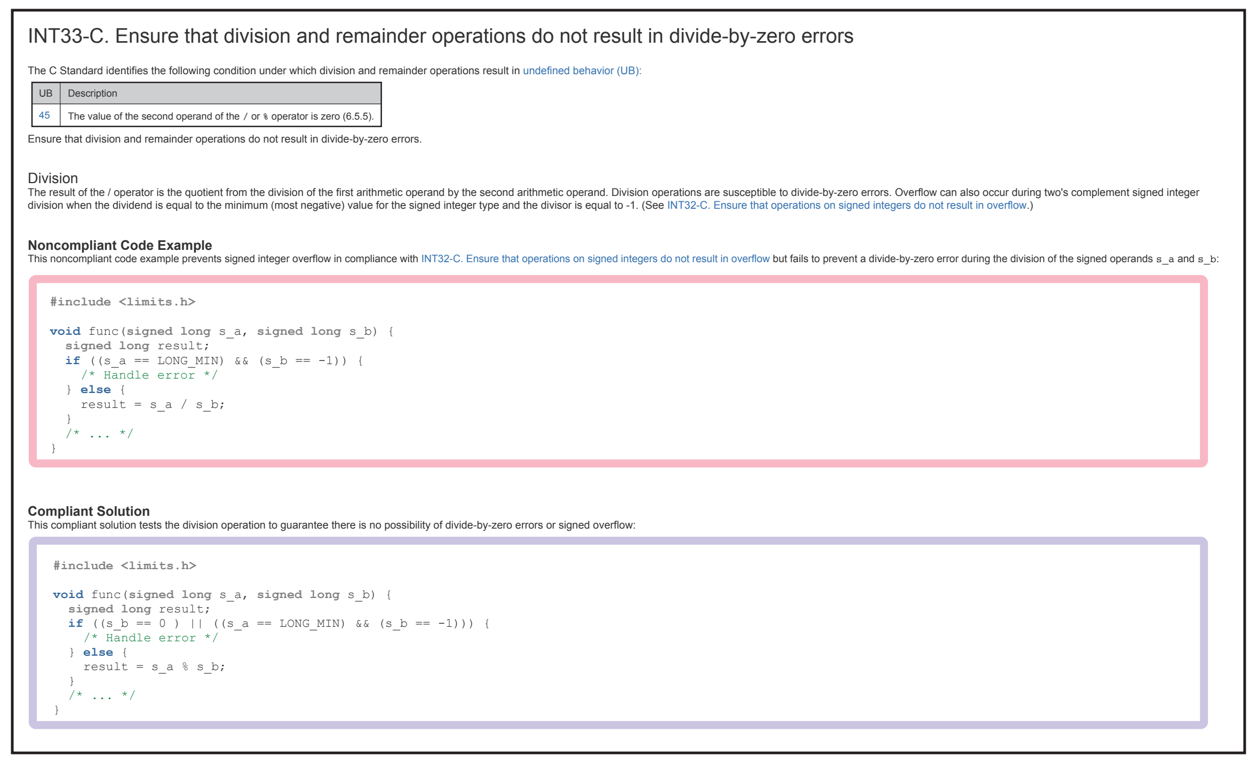
Team:

- 2 graduate students
- 2 undergraduate students
- 3-5 SwA experts
- No IBM Watson experience
- Used Python and JSON interfaces
- 11 weeks

*https://en.wikipedia.org/wiki/Watson_(computer)

Example original document: CERT INT33-C Rule - Parts

- IBM Watson works on Solr document
- Each rule or CWE resulted in about 11 Solr documents
- Whole rule or CWE is a Solr document
- Key sections are Solr documents
- Many different formats within document
- Corpus held about 15,000 documents



INT33-C. Ensure that division and remainder operations do not result in divide-by-zero errors

The C Standard identifies the following condition under which division and remainder operations result in undefined behavior (UB):

UB	Description
45	The value of the second operand of the / or % operator is zero (6.5.5).

Ensure that division and remainder operations do not result in divide-by-zero errors.

Division

The result of the / operator is the quotient from the division of the first arithmetic operand by the second arithmetic operand. Division operations are susceptible to divide-by-zero errors. Overflow can also occur during two's complement signed integer division when the dividend is equal to the minimum (most negative) value for the signed integer type and the divisor is equal to -1. (See INT32-C. Ensure that operations on signed integers do not result in overflow.)

Noncompliant Code Example

This noncompliant code example prevents signed integer overflow in compliance with INT32-C. Ensure that operations on signed integers do not result in overflow but fails to prevent a divide-by-zero error during the division of the signed operands s_a and s_b:

```
#include <limits.h>

void func(signed long s_a, signed long s_b) {
  signed long result;
  if ((s_a == LONG_MIN) && (s_b == -1)) {
    /* Handle error */
  } else {
    result = s_a / s_b;
  }
  /* ... */
}
```

Compliant Solution

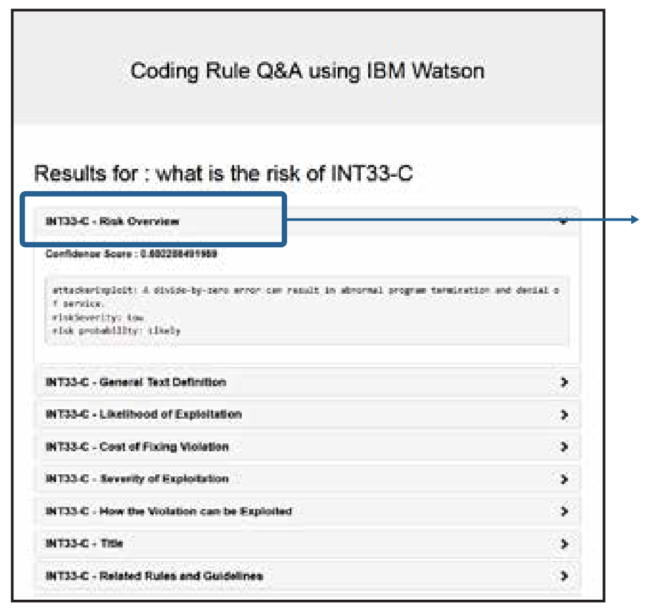
This compliant solution tests the division operation to guarantee there is no possibility of divide-by-zero errors or signed overflow:

```
#include <limits.h>

void func(signed long s_a, signed long s_b) {
  signed long result;
  if ((s_b == 0 ) || ((s_a == LONG_MIN) && (s_b == -1))) {
    /* Handle error */
  } else {
    result = s_a % s_b;
  }
  /* ... */
}
```

Application performance

Better Recall and Precision: Example: "What is the risk of INT33-C"



Coding Rule Q&A using IBM Watson

Results for : what is the risk of INT33-C

INT33-C - Risk Overview

Confidence Score : 0.600228491989

```
attackerExploit: A divide-by-zero error can result in abnormal program termination and denial o
f service.
riskSeverity: low
risk probability: likely
```

INT33-C - General Text Definition

INT33-C - Likelihood of Exploitation

INT33-C - Cost of Fixing Violation

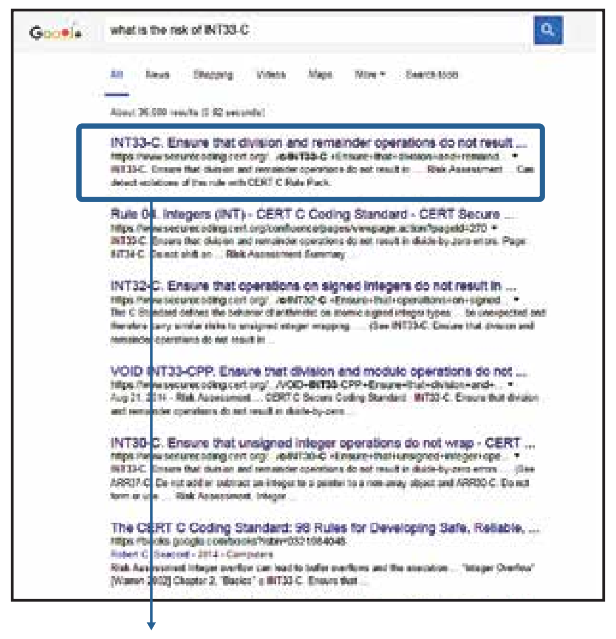
INT33-C - Severity of Exploitation

INT33-C - How the Violation can be Exploited

INT33-C - Title

INT33-C - Related Rules and Guidelines

INT33-C – Risk Overview



what is the risk of INT33-C

About 36,000 results (0.62 seconds)

INT33-C. Ensure that division and remainder operations do not result ...
https://www.securecoding.cert.org/.../c/INT33-C.+Ensure+that+division+and+remaind...
INT33-C. Ensure that division and remainder operations do not result in ... Risk Assessment ... Can detect violations of this rule with CERT C Rule Pack.

Rule 04. Integers (INT) - CERT C Coding Standard - CERT Secure ...
https://www.securecoding.cert.org/confluence/pages/viewpage.action?pageId=270
INT33-C. Ensure that division and remainder operations do not result in divide-by-zero errors. Page INT34-C. Do not shift an ... Risk Assessment Summary.

INT32-C. Ensure that operations on signed integers do not result in ...
https://www.securecoding.cert.org/.../c/INT32-C.+Ensure+that+operations+on+signed ...
The C Standard defines the behavior of arithmetic on atomic signed integer types ... be unexpected and therefore carry similar risks to unsigned integer wrapping ... (See INT33-C. Ensure that division and remainder operations do not result in ...

VOID INT33-CPP. Ensure that division and modulo operations do not ...
https://www.securecoding.cert.org/.../VOID-INT33-CPP.+Ensure+that+division+and+...
Aug 21, 2014 - Risk Assessment ... CERT C Secure Coding Standard: INT33-C. Ensure that division and remainder operations do not result in divide-by-zero ...

INT30-C. Ensure that unsigned integer operations do not wrap - CERT ...
https://www.securecoding.cert.org/.../c/INT30-C.+Ensure+that+unsigned+integer+ope...
INT33-C. Ensure that division and remainder operations do not result in divide-by-zero errors ... (See ARR37-C. Do not add or subtract an integer to a pointer to a non-array object and ARR30-C. Do not form or use ... Risk Assessment. Integer ...

The CERT C Coding Standard: 98 Rules for Developing Safe, Reliable, ...
https://books.google.com/books?isbn=0321984048
Robert C. Seacord - 2014 - Computers
Risk Assessment. Integer overflow can lead to buffer overflows and the execution ... "Integer Overflow" [Warren 2002] Chapter 2, "Basics" ... INT33-C. Ensure that ...

INTC33-C. Ensure that division and remainder operations do not result …
https://www.securecoding.cert.org/…/c/INT33-C. =Ensure+that+dividion+and+remaind…

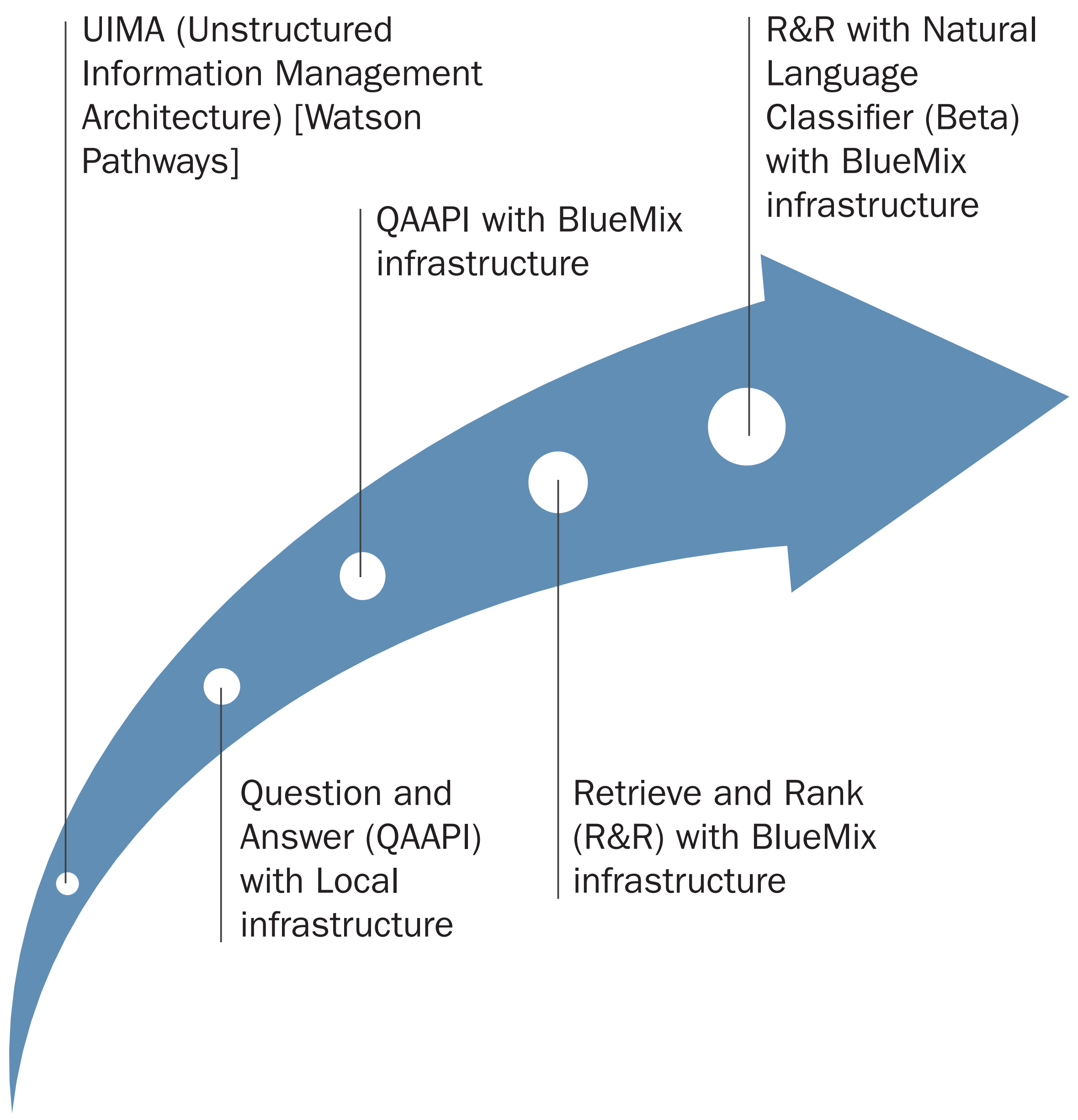
Watson's interfaces for cognitive querying evolved over time

Organization of technology rapidly evolved

- Splitting some components into distinct services
- Combining some services into usable chunks
- Ease-of-use interfaces delivered in open source (out of product cycle)

Project focused on using "Retrieve and Rank" on BlueMix

- Available support from IBM
- Combined Watson Pathways for Concept Expansion, Concept Insights and Question-and-Answer


UIMA (Unstructured Information Management Architecture) [Watson Pathways]
Question and Answer (QAAPI) with Local infrastructure
QAAPI with BlueMix infrastructure
Retrieve and Rank (R&R) with BlueMix infrastructure
R&R with Natural Language Classifier (Beta) with BlueMix infrastructure

Lessons learned from project

Theory	Practice
Automated natural language comprehension	SME-driven Q&A training

Training uses about 150,000 questions and answers

Disposition of materials

Government use rights apply. IBM Watson software (and any dependencies) must be licensed from IBM.

SparkCognition is an IBM Watson business partner (independent software vendor) and has licensed the project materials from CMU for use in their products.

We want to thank and acknowledge collaborators

SparkSecure team at SparkCognition

IBM Watson team at IBM

Prof. Eric Nyberg, Language Technologies Institute, School of Computer Science, CMU

And our student interns: Christine Baek, Anire Bowman, Skye Toor and Myles Blodnick

Distribution Statement A: Approved for Public Release;
Distribution is Unlimited

Contact: Mark Sherman | mssherman@cert.org