Some of the work described in this talk was funded by the US National Science Foundation under Grants CCF-0959924 and CCF-1265178.
Business intelligence improves decision making, cuts costs, and is used to identify new business opportunities. It leverages data mining, analytical processing, querying and reporting.

We coin the term **PROJECT intelligence** and apply it to the data produced – and decisions made, during the Software Development Life Cycle.
Project data

- Regulatory Codes
- System Requirements
- Sub-System Requirements
- Sub-System Requirements
- Component Requirements
- Component Requirements
- Stakeholders

Source Code in packages, components, and versions.

Fault Logs

<table>
<thead>
<tr>
<th>Error / Warning Message</th>
<th>Description of Error / Warning</th>
<th>Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot override existing file</td>
<td>Selected filename is read-only.</td>
<td>Select another filename and overwrite.</td>
</tr>
<tr>
<td>Could not open &lt;filename&gt; file to display</td>
<td>Error occurred if the selected configuration or selection file cannot be retrieved. If the user issues help and there is no help file,</td>
<td>Correct spelling of filename when prompted for filename to retrieve.</td>
</tr>
<tr>
<td>Missing required deleted &lt;filename&gt; compound code</td>
<td>SSP code specified is not on the database.</td>
<td>Under &quot;Run autoBBSP&quot;, enter correct SSP compound code.</td>
</tr>
<tr>
<td>Could not open &lt;filename&gt;</td>
<td>Filename is to be removed cannot be opened.</td>
<td>Check path name and filename spelling. Check for permissions.</td>
</tr>
<tr>
<td>Unknown field &lt;filename&gt; must be disabled or enabled</td>
<td>In configuration file for the options for the source file are either disabled or enabled.</td>
<td>On the command line of the UMS shell, either the configuration file or type either Disabled or Enabled for the fault log.</td>
</tr>
<tr>
<td>XENVIRONMENT not set</td>
<td>Environment variable not set up, and control is passed to the OUI's auxiliary file.</td>
<td>Limit environment variable setup.</td>
</tr>
</tbody>
</table>

Stakeholders

- Source Code in packages, components, and versions.

Change Requests

- Deployment
- Change Request Management
- Internal Test Release
- Implementation Bug Processing

High Level Activity

<table>
<thead>
<tr>
<th>Timeline</th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and resourcing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Needs Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtual Desktop support process development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architect &amp; Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Lab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Acceptance Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production Build</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Installation of user Virtual Desktops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-installation support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User feedback &amp; lessons learned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operations handover</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Project Plans

- Change Requests
- Documentation
- Sub-System Requirements
- Component Requirements
- User Requirements
- System Requirements
- Regulatory Codes

Acceptance Tests

NO YES
Project Intelligence: Challenges

Distributed data

Data connectivity

Heterogeneous data

SELECT DISTINCT regulatory_code.*
FROM ((regulatory_code
    LEFT OUTER JOIN tm_regulatory_code2software_requirement
    ON (regulatory_code.id = tm_regulatory_code2software_requirement.software_requirements_id))
    LEFT OUTER JOIN software_requirement
    ON (software_requirement.id = tm_regulatory_code2software_requirement.software_requirement_id)
WHERE regulatory_code.relevant = 'true'
AND software_requirement.id IS NULL;
Software data can be used strategically to deliver meaningful project intelligence.
Webinar Goals

Suggest **Strategic questions** that could be addressed by project data.

Identify **Analytic** queries that deliver critical Project Intelligence.

**Instrument** your development environment to connect data.

Explore cutting edge **Research** that aims at enabling project intelligence.

Embrace **Traceability** as the new **black**!
Which regulatory codes are relevant to my system?

Are all relevant regulatory codes fully addressed in the as-built system?

How do changes in the updated codes impact my system?
Queries: Project Test Status

- What percentage of acceptance tests have passed?
- Should we invest further time testing?
- Have any components repeatedly failed acceptance tests?
- Do any of the reused components have a history of post-release faults?
- Who is a domain expert for this problematic component?
Queries: Bug Triage and Fixing

Source Code

Bug Reports and Change Requests

What is the history of changes for this component?

Who should we assign this bug to?

Who are the experts on the interface to HADOOP?

Are any new fault trends emerging that we haven’t previously seen? How critical are they?

Programmers

Bug fix performance history
Queries: Project Management

Are there any bottlenecks in the continuous integration?

How well are we performing on the current project?

How did the recent process change impact the delivery stream?
What questions do you have?

- Safety
- Compliance
- Process
- ROI
- Code Quality
- Architecture
- Project
- Triage
- Tests
- Quality
- Faults
- Project
- ROI
- Architecture
- Code Quality
- Architecture
A Closer Look at the Challenges

Distributed data

Heterogeneous data

Data connectivity

Necessary support for Software Analytics

SELECT DISTINCT regulatory_code.*
FROM ((regulatory_code LEFT OUTER JOIN tm_regulatory_code2software_requirement ON (regulatory_code.id = tm_regulatory_code2software_requirement.software_requirements_id)) LEFT OUTER JOIN software_requirement ON (software_requirement.id IS NULL)
WHERE regulatory_code.relevant='true' AND software_requirement.id IS NULL;
Challenge 1: Accessing & Integrating Data

Unless your connectivity goals are quite limited consider using a tool or service which provides data connectivity between software artifacts.
Challenge 2: Knowing what you want

What?

Why?

How?

Who_1?

Who_2?

When?

Preliminary Hazard
- ID
- riskLevel {high/medium/low}
- description

Prevents

System Requirement
- ID
- description

Derives from

Regulatory Code
- ID
- name
- description
- relevant

Complies to

Software Requirement
- ID
- description
- type

Tests

Acceptance Test
- ID
- name
- result {failed/passed}
- description
- date modified
- date created

Tests

Code
- ID
- name
- createDate
- Version
- description

Tests

Unit Test
- ID
- name
- description
- result {failed/passed}
- testDate

Documents results

Test Log
- ID
- expectedResults
- actualResults
- dateRun
- comments

Documents results

Unit Test Result
- ID
- resultStatus {failed/passed}
- dateRun

Documents results
Challenge 3: Connecting the Artifacts

The ability to **interrelate any uniquely identifiable** software engineering artifact to any other, **maintain** required links over time, and **use the resulting network** to answer questions of both the software product and its development process.

- CoEST Definition
Tracing Mistakes People Make

1. Failure to plan: Most projects don’t include a TIM
2. Ill-defined trace granularity leads to unacceptably high/low or mismatched links.
3. Redundant Trace Paths
4. Lack of unique project-wide IDs
5. Missing trace links
6. The DOORS dump – i.e. massive unstructured dumps of trace data
7. Traceability as an after-thought.
Mind the Gap: Assessing the Conformance of Software Traceability to Relevant Guidelines, Patrick Rempel, Patrick Mäder, Tobias Kuschke (TU Ilmenau), and Jane Cleland-Huang (DePaul), ICSE 2014, Hyderabad, India
Hazard H2: Moving the patient’s arm at excessive velocity.

Fault F2.1: Velocity sensors fail to sense excessive velocity.

Fault F2.2: Configuration component fails to update correct velocity constraints.

R1: A system test must be run prior to each use to check that the sensors are operating correctly.

R2: All sensors must be duplicated.

R3: The robotic arm must stop automatically if arm sensors disagree on current velocity by more than x mps.

R9: Current velocity constraint is displayed on the monitor.

R10: Movement must be disabled if current velocity constraint fails to match patient’s personal record.

R11: Current velocity constraint must fall under maximum allowed velocity.

Test Case T1
Test Case T2
Test Case T3
Test Case T4
Test Case T5
Test Case T6
Test Case T7
Test Case T8

Includes trace links created using Requirements management tools such as DOORS.
Automated approaches leverage embedded text to compute similarity between artifacts.

Poirot: DePaul University

RETO: University of Kentucky
Towards more Intelligent Traceability

What goes through a domain expert’s mind as he or she performs the tracing task?
Human Analysts think about concepts...

Status of field elements is consumed by the **WIU**, which in turn creates a **wayside status message** and broadcasts that message out to any **automobile** within range.

The **Highway Wayside Segment** shall transmit information to the **automobile controller** in the form of **WSMs**.

**WIU** = Wayside Interface Unit and is located in a Highway Wayside Segment.

Broadcast is similar to transmit.

Automobile controller is part of the automobile

WSM and wayside status message are equivalent.

Both artifacts involve Highway Wayside Segment sending wayside status message to automobiles.

How do I trace this?
Domain Centric Expert Traceability

1. Identify **Verbs**.
2. Categorize each verb by its **usage group**.
3. Identify **nouns and noun phrases** associated with each verb. Assign **thematic roles** to each noun and noun phrase.
4. Identify the **semantic group** of each action unit’s verb.
5. Apply **heuristics** to pairs of action units across source and target artifacts to determine whether a trace link exists. **Create trace links accordingly.**

An Action Frame is normally defined by:
- A **verb**
- A **semantic type**.
- Nouns & noun phrases assigned **thematic roles**.

Domain Ontology Link Heuristics

Jin Guo  
Natawut Monaikul  
Cody Plepel
A1: The OBM shall support reception and decomposition of Wayside Status Messages.

**Action**: reception

**Semantic Groups**: Receptive

**Recipients**: OBM

**Themes**: wayside status message

A2: The Wayside Segment shall transmit information to the Automobile Segment in the form of Wayside Status Messages.

**Action**: transmit

**Semantic Groups**: Transmissive

**Agents**: wayside segment

**Recipients**: automobile segment

**Themes**: wayside status message

1. **Semantic group** in one action group is **Transmissive** and the other is **receptive**. (Applicable)

2. Compare **Agents**, **Recipients**, **Instruments**, **Locations**. If a pair is present, it must exhibit an exact or hierarchical match. If only one side is present, this step is skipped.

3. Compare **Themes**.

Both sides of a pair must be present and they must exhibit an exact or hierarchical match.
97% of action units extracted correctly from SRS
76.5% of action units extracted correctly from SDD

Very significant improvement in both recall and precision when DoCIT is used.

This is just our first prototype. We believe we can do much better.
Monitor the project environment and use **project exhaust** to infer trace links based on sequencing of tasks and version control check-in information.

Source code changes committed to the repository are tagged with references to test cases and requirements.

The project environment is monitored and artifacts worked on simultaneously are tentatively associated.
The process requires artifacts to be created in certain ways – enforcing trace link creation.
Evolving Trace Links

Evolve trace links dynamically as requirements, code, and other artifacts change over time.
How accessible is the data?

Even when trace links exist, it can be challenging to retrieve, and extract the data in meaningful ways.
Challenge 4: Querying the Data

Return all code classes which are associated with high risk preliminary hazards and which have failed test cases in the past month.
Formulating Typical Project Queries

One of the primary barriers to leveraging project data in practice is the difficulty of using previously created trace links.

Insufficient tool support.

---

Subject: traceability query

Hey all I need a super query writer... I have the following query:

```
SELECT req rq_user_17 AS "Requirement Type", REQ.RQ_REQ_ID as "Requirement ID", REQ.RQ_USER_08 as "Requirement Class", REQ.RQ REQ_NAME as "Requirement Name", count (REQ COVER.RC ITEM_ID) AS "Coverage"
FROM REQ, req_cover
WHERE REQ.RQ_REQ_ID in (select REQ COVER.RC REQ_ID from
req_cover)
and req rq_user_17=WS'
GROUP BY req rq_user_17, REQ.RQ_REQ_ID,
REQ.RQ_USER_08, REQ.RQ_REQ_NAME
union
SELECT req rq_user_17 AS "Requirement Type", REQ.RQ_REQ_ID as "Requirement ID", REQ.RQ_USER_08 as "Requirement Class", REQ.RQ REQ_NAME as "Requirement Name", count (REQ rq_attachment) AS "Coverage"
FROM REQ, req_cover
WHERE REQ.RQ_REQ_ID not in (select REQ COVER.RC REQ_ID from
req_cover)
and req rq_user_17=WS'
GROUP BY req rq_user_17, REQ.RQ_REQ_ID,
REQ.RQ_USER_08, REQ.RQ_REQ_NAME
order by req rq_user_17
```

It gives me the aggregate number of testcase in req_cover and it puts this number to each requirement. But what I need is the aggregate number of testcase linked to each requirement. can anyone help?
Query Solutions

SQL (with or without query builders)

SELECT DISTINCT regulatory_code.*
FROM (regulatory_code
    LEFT OUTER JOIN tm_regulatory_code2software_requirement
    ON regulatory_code.id = tm_regulatory_code2software_requirement_id)
LEFT OUTER JOIN software_requirement
ON (software_requirement.it = tm_regulatory_code2software_requirement.software_requirements_id)
WHERE regulatory_code.relevant = 'true'
AND software_requirement.id IS NULL;

Visual Queries

Give me a list of all java classes associated with high risk preliminary hazards which have failed test cases in the past month.
Towards Natural Language Queries

**Ask Jeeves**: Smart Answers Fast!

Just type a question and click “Ask!”
For example: How old am I?

**Ask Jeeves**

What percentage of acceptance tests have passed?

Related Search:
- Acceptance Quotes
- Role of a User Acceptance Testing
- Acceptance Testing Checklist
- Acceptance Letter Example
- Definition of Acceptance
- Essay Samples for Admission in Nursing Program

User Acceptance Test Plan Templates
Sample of a Letter of Acceptance for a New Job
Self Acceptance
Regional Acceptance Corporation
Social Acceptance
Proper Way to Write a RSVP Response

Web Results:
- **Test-driven development**
  en.wikipedia.org/wiki/Test-driven_development
  By focusing on writing only the code necessary to pass tests, designs can often be... the development team now has a specific target to satisfy – the acceptance tests. ... The common practice...
- **SSW - Rules to better User Acceptance Tests (UAT) - Sydney...**
  Figure: A User Story with Acceptance Criteria (MSF Agile Template) ... approach, you do not have historical data for your test pass rates for previous milestones.
- **Possible Metrics for Acceptance Testing to Evaluate Vendor...**
  Dec 6, 2002 ... Although there are numerous metrics for software testing, and new ones ... Success rate—The percentage of test cases that passed at the last execution was... in addition to vendor-delivered software, for projects that have the...
More Siri fun!

OK, I found this on the web for 'Who is a domain expert for this problematic component':

BING SEARCH

Subject-matter expert - Wikipedia, the free encyclopedia
... or domain expert is a person who is an authority in a ... According to Six Sigma, a Subject Matter Expert "exhibits the highest le...
en.wikipedia.org

Chp 1: Expert Systems And Artificial Intelligence
Chapter 1 INTRODUCTION. Robert S. Enzelmare Edward Feigenbaum. EXPFRT

"Do any of the reused components have a history of post release faults"
tap to edit

Interesting question.
TiQi allows a user to express a trace query in spoken or written natural language and then transforms it into executable SQL statements.

- Are there any hazards with no identified contributing faults?
- List all tests which have recently failed and which are associated with high severity faults.
- List all requirements related to heat sensor faults.
- Is the system safe for use?
- What’s up dude?
I’d like to see a list of all preliminary hazards for arm movements which are tested by recent unit tests.

```sql
SELECT PreliminaryHazard.*
FROM PreliminaryHazard, LINKSystemRequirement2PreliminaryHazard, SystemRequirement,
LINKSoftwareRequirement2SystemRequirement, SoftwareRequirement,
LINKUMLClass2SoftwareRequirement, UMLClass,
LINKCodeClass2UMLClass, UMLCode,
LINKUnitTest2CodeClass, UnitTest
WHERE PreliminaryHazard.ID = LINKSystemRequirement2PreliminaryHazard.TargetID AND
SystemRequirement.ID = LINKSystemRequirement2PreliminaryHazard.SourceID AND
SoftwareRequirementID.ID = LINKSoftwareRequirement2SystemRequirement.TargetID AND
SoftwareRequirement.ID = LINKSoftwareRequirement2SystemRequirement.SourceID AND
SoftwareRequirement2SoftwareRequirement.ID = LINKUMLClass2SoftwareRequirement.TargetID AND
UMLClass.ID = LINKUMLClass2SoftwareRequirement.SourceID AND
UMLCode.ID = LINKCodeClass2UMLClass AND
UnitTest.ID = LINKUnitTest2CodeClass AND
UnitTest.testDate >= '03/01/2014' AND PreliminaryHazard.Description LIKE '%arm movement%';
```
Transforming NL to SQL

**SQL Query**
```
SELECT DISTINCT `software-requirements`.`ID`, `software-requirements`.`description` 
FROM `software-requirements`, `uml-class`, `code-class`, `tm_codeuml`, `tm_umlsrs` 
WHERE (`tm_umlsrs`.`SrsID` = `software-requirements`.`ID` AND `tm_umlsrs`.`umlID` = `uml-class`.`ID` AND `tm_codeuml`.`umlID` = `uml-class`.`ID` AND `tm_codeuml`.`CodeID` = `code-class`.`ID`) 
```

**Query Output**

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS1</td>
<td>Input laboratory Registry - It should allow the operator to enter the Registry of a laboratory analysis</td>
</tr>
<tr>
<td>SRS2</td>
<td>Changing laboratory Registry - It allows the operator to change the Registry of a laboratory analysis. It deals with the modification of data about the Registry of laboratory analysis</td>
</tr>
</tbody>
</table>

**VTML Query**

**NLQ:** Show all software requirements realized in UML classes but not yet implemented in code

Queries involve Different Data Types

1. Retrieve all requirements with open issues. **Raw Data**
2. Show trends in fault occurrence in the past 2 years. **Synthesized Data**
3. Predict which safety-critical components are likely to exhibit future bugs. **Produced Data**
We can include software analytics as attributes available for querying.
Software Analytics: It's just data!

① Model Features and their valid compositions

② Implement as components

③ Configure analytic functions.

④ Instrument the Query Environment (SQL, VTML, TiQi)

Source Code

Class Name
Code
CreatedBy
Created On

f Fault prone
f Find-the-expert
f Topic
And finally... An analytics query

Who is the best person to check the fault prone classes in components related to temperature controls?
Next Steps

① Identify useful Queries

② Identify relevant artifacts and determine who, what, when, why? Build a model as a strategic plan.

③ Instrument the environment to capture, create, or infer trace links.

④ Connect your data

⑤ Design useful analytic functions and configuring the environment accordingly.
Why is Traceability the new black?

Project intelligence delivers real value..

Traceability empowers project intelligence.

The cost of connecting (tracing) artifacts continues to decrease....
Traceability is Coming Soon. For further information visit CoEST.org

ACM Webinar, October 7th 2015

Jane Cleland-Huang, PhD
jhuang@cs.depaul.edu
Center of Excellence for Software Traceability
DePaul University

Moderated by:
Sarah Gregory
Intel Corporation

Some of the work described in this talk was funded by the US National Science Foundation under Grants CCF-0959924 and CCF-1265178.