

# Projekt Themen

# Übersicht

1. What Information Does a Developer Need?
2. Freeze Mental Model to Generating Tests
3. Expertise Search Interface
4. Social Coding
5. Query Explanation
6. Analytical User Interface based on a Set-based Query Language for HYRISE

# 1. What Information Does a Developer Need?



## 2. Freeze Mental Model to Generating Tests

File
Edit
Tools

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

wire transfer (
  IN from_account INT
  IN to_account INT
  IN trans_amount INT
  OUT new_balance INT
  OUT success BOOL)

current_balance := SELECT SUM(amount) FROM transfers
WHERE acct = from_account

IF current_balance - trans_amount >= 0 THEN
  INSERT INTO transfers(acct, amount)
  VALUES
    (from_account, -trans_amount),
    (to_account, trans_amount)
  new_balance := current_balance - trans_amount - 1000
  success := TRUE
ELSE
  new_balance := current_balance
  success := FALSE
END IF

```

**Test Generator**

SETUP

from\_account = 3  
to\_account = 1  
trans\_amount = 1000

---

Used tables (Test Data Set)

Transfer

acct	amount
3	5000
3	6000

---

EXPECTED

new\_balance = 10000  
success = TRUE

Side effects

INSERT: (3, -1000)  
into Transfer

INSERT: (1, 1000)  
into Transfer

<http://vimeo.com/36579366>, min 16:48

# 3. Expertise Search Interface

- *How to find existing expertise about a certain source code method without having to leave the IDE?*
- Development of an Eclipse plugin that allows me to
  - Select a method from my source code
  - Display relevant contacts and artifacts (wiki pages, emails, tweets, ...)
  - Store the search history and use it in subsequent requests
- Prototype should be validated in small user tests
- What you learn: Eclipse plugin development, HANA, User testing

# Mockups

application.coffee

```

class Observable
  constructor: ->
    @listeners = {}
    @events = []

  on: (name, callback) ->
    if name not in @events
      return

    if name not in @listeners
      @listeners[name] = []

    @listeners[name].push callback

  raise: (name) ->
    args = ($ arguments).slice 1

    callback.apply @, args for callba

  un: (name, callback) ->
    @listeners[name].pop callback if
  
```

- 7.0 Discussion (Hans and Peter)  
3 Messages (1. May 2012)
- 3.0 Wiki (Event handling)  
Author: Hans (5. May 2012)
- 2.0 Expert (Peter)  
Honoured on 25. June 2012

Das Bild kann nicht angezeigt werden. Dieser Computer verfügt möglicherweise über wenig Arbeitsspeicher, um das Bild zu öffnen, oder das Bild ist beschädigt. Starten Sie den Computer neu, und öffnen Sie dann erneut die Datei. Wenn weiterhin das rote x angezeigt wird, müssen Sie das Bild möglicherweise löschen und dann erneut einfügen.

application.coffee

```

class Observable
  constructor: ->
    @listeners = {}
    @events = []

  on: (name, callback) ->
    if name not in @events
      return

    if name not in @listeners
      @listeners[name] = []

    @listeners[name].push callback

  raise: (name) ->
    args = ($ arguments).slice 1

    callback.apply @, args for callba

  un: (name, callback) ->
    @listeners[name].pop callback if
  
```

- 7.0 Discussion (Hans and Peter)  
3 Messages (1. May 2012)
- 3.0 Wiki (Event handling)  
Author: Hans (5. May 2012)
- 2.0 Expert (Peter)  
Honoured on 25. June 2012

### Event Handling

Author: Peter  
Last changed: 10/03/2012

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui ...

application.coffee

```

class Observable
  constructor: ->
    @listeners = {}
    @events = []

  on: (name, callback) ->
    if name not in @events
      return

    if name not in @listeners
      @listeners[name] = []

    @listeners[name].push callback

  raise: (name) ->
    args = ($ arguments).slice 1

    callback.apply @, args for callba

  un: (name, callback) ->
    @listeners[name].pop callback if
  
```

- 7.0 Discussion (Hans and Peter)  
3 Messages (1. May 2012)
- 3.0 Wiki (Event handling)  
Author: Hans (5. May 2012)
- 2.0 Expert (Peter)  
Honoured on 25. June 2012

**Peter**  
Senior Developer  
Tel: 1234-56789  
IM: Han Solo

### Expert History

01/01/2012 - edit  
12/12/2011 - create

### Schedule

08:00	.....	...
09:00	.....	...
10:00	.....	...
11:00	.....	Meeting
12:00	.....	...

# Topic Outline – Master Seminar

- *How to identify expertise about source code methods?*
- Research & Implementation Tasks:
  - Overview about existing expert(ise) detection systems and their employed algorithms
  - Prototypical implementation of a real-time expert detection system using SAP HANA
    - Storage of artifact and expert information
    - Storage of search histories
    - Creation of queries/stored procedures that help determining potentially interesting documents and people
  - Evaluation of the prototype by performing small user tests

# Starting Points

- Master's thesis: Ralf Gehrler, Topic detection in heterogenous groupware landscapes
- Girish Maskeri, Santonu Sarkar, and Kenneth Heafield. Mining business topics in source code using latent dirichlet allocation. In *Proceedings of the 1st India software engineering conference, ISEC '08*, pages 113–120, New York, NY, USA, 2008. ACM.
- Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre Baldi. Mining concepts from code with probabilistic topic models. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, ASE '07*, pages 461– 464, New York, NY, USA, 2007. ACM.
- ...



# 4. Social Coding: Code Annotations

- Code documentation is formal
  - Description of current status (or should be status)
    - no place for questions
- Code documentation is rather static
  - Mental barrier to change it is rather high
  - No „organic“ co-evolution with code
- Wanted improvements
  - Informal possibility to
    - ask questions
    - give feedback
    - rant about bad code
  - Your ideas are welcome, too!



# Social Coding: Code Annotations

- Facebook for Code

- Let us develop an Eclipse based prototype to test ideas

- Multiple usage scenarios possible

- Improvement of documentation
- Identification of interesting code pieces
- Enhanced knowledge transfer

- Work packages

- Usage Concept
- Vertical Prototype
- UI Design (pen&paper, please!)
- Implementation

- You will get insights into

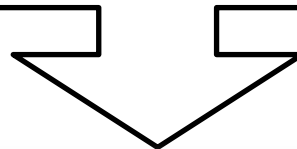
- Eclipse development
- git connectivity
- HANA

# 5. Query Explanation

The SQL EXPLAIN command is used to display the execution plan of an SQL query.

Example:

```
EXPLAIN SELECT DISTINCT A.date, A.volume
FROM EDEKA.SALES JOIN AS A
  LEFT OUTER JOIN EDEKA.PROMOTIONS JOIN AS F on A.JOIN_KEY = F.JOIN_KEY AND F.BEFID = 'A99'
  LEFT OUTER JOIN EDEKA.IGNORES JOIN AS I ON A.JOIN_KEY = I.JOIN_KEY
  LEFT OUTER JOIN EDEKA.DATESJOIN AS K99 ON A.JOIN_KEY = K99.JOIN_KEY AND K99.BEFID = 'K99'
  LEFT OUTER JOIN EDEKA.PRICINGJOIN AS P ON A.JOIN_KEY = P.JOIN_KEY AND P.BEFID = 'SPR'
WHERE P.price IS NOT NULL
  AND I.ignored IS NULL
  AND K99.dated IS NULL
  AND F.promoted IS NULL
  AND A.warehouse_id=3
  AND A.product_id=461341
ORDER BY A.date
```

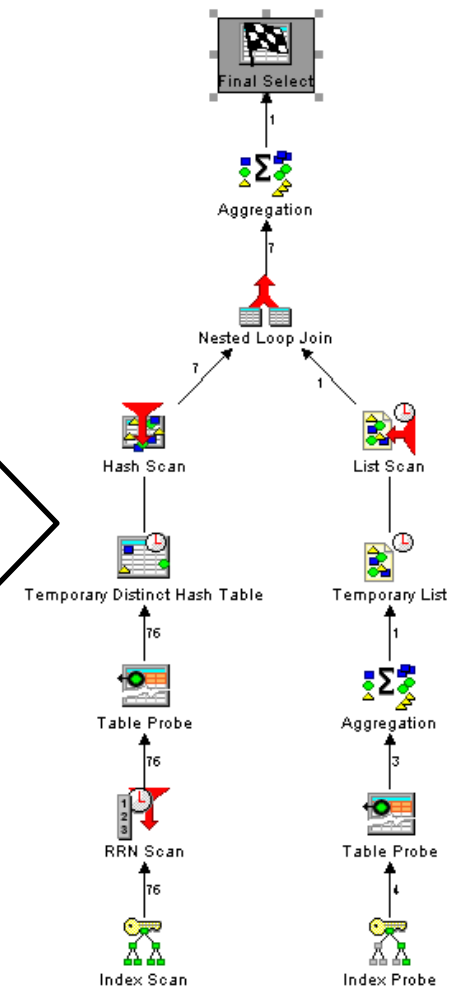


STATEMENT_NAME	OPERATOR_NAME	OPERATOR_DETAILS	EXECUTION_ENGINE	SCHEMA_NAME	TABLE_NAME	TABLE_TYPE	TABLE_SIZE	OUTPUT_SIZE	SUBTREE_COST	OPERATOR_ID	PARENT
1	?	COLUMN SEARCH	A.DATE, A.VOLUME (LATE MATERIALIZATION)	COLUMN	?	?	?	?	1 0.07255841705061498	1	
2	?	ORDER BY	A.DATE ASC	COLUMN	?	?	?	?	1 0.07255651204473955	2	
3	?	DISTINCT	GROUPING: A.DATE, A.VOLUME	COLUMN	?	?	?	0.950000...	0.07255651204473955	3	
4	?	JOIN	JOIN CONDITION: (INNER) AJOIN_KEY = P.J...	COLUMN	?	?	?	?	1 0.07255552040573145	4	
5	?	COLUMN...	FILTER CONDITION: P.BEFID = 'SPR' AND P...	COLUMN	EDEKA	PRICINGJOIN	COLUMN...	37,677,...	37,677,692	?	5
6	?	COLUMN...	A.DATE, A.VOLUME, AJOIN_KEY	COLUMN	?	?	?	?	0.000026...	0.07026204767880267	6
7	?	DISTINCT	GROUPING: A.DATE, A.VOLUME, AJOIN_KEY	COLUMN	?	?	?	?	0.000026...	0.06611640988286775	7
8	?	FILTER	I.IGNORED IS NULL AND F.PROMOTED IS N...	COLUMN	?	?	?	?	1 0.06611617429652995	8	
9	?	JOIN	JOIN CONDITION: (LEFT OUTER) AJOIN_KE...	COLUMN	?	?	?	?	15,766.13...	0.06611591133405452	9
10	?	COL...	FILTER CONDITION: A.WAREHOUSE_ID = 3...	COLUMN	EDEKA	SALESJOIN	COLUMN...	13,889,...	13,889.962	?	10
11	?	COL...	FILTER CONDITION: F.BEFID = 'A99'	COLUMN	EDEKA	PROMOTI...	COLUMN...	3,162,718	15,813.59	?	11
12	?	COL...		COLUMN	EDEKA	IGNORESJOIN	COLUMN...	280,017	280,017	?	12
13	?	COL...	FILTER CONDITION: K99.BEFID = 'K99'	COLUMN	EDEKA	DATESJOIN	COLUMN...	2,068,514	2,068.514	?	13

# Query Explanation - Reloaded

```

EXPLAIN SELECT DISTINCT A.date, A.volume
FROM EDEKA.SALES JOIN AS A
  LEFT OUTER JOIN EDEKA.PROMOTIONS JOIN AS F on A.JOIN_KEY = F.JOIN_KEY
    AND F.BEFID = 'A99'
  LEFT OUTER JOIN EDEKA.IGNORES JOIN AS I ON A.JOIN_KEY = I.JOIN_KEY
  LEFT OUTER JOIN EDEKA.DATESJOIN AS K99 ON A.JOIN_KEY = K99.JOIN_KEY
    AND K99.BEFID = 'K99'
  LEFT OUTER JOIN EDEKA.PRICINGJOIN AS P ON A.JOIN_KEY = P.JOIN_KEY
    AND P.BEFID = 'SPR'
WHERE P.price IS NOT NULL
  AND I.ignored IS NULL
  AND K99.dated IS NULL
  AND F.promoted IS NULL
  AND A.warehouse_id=3
  AND A.product_id=461341
ORDER BY A.date
    
```

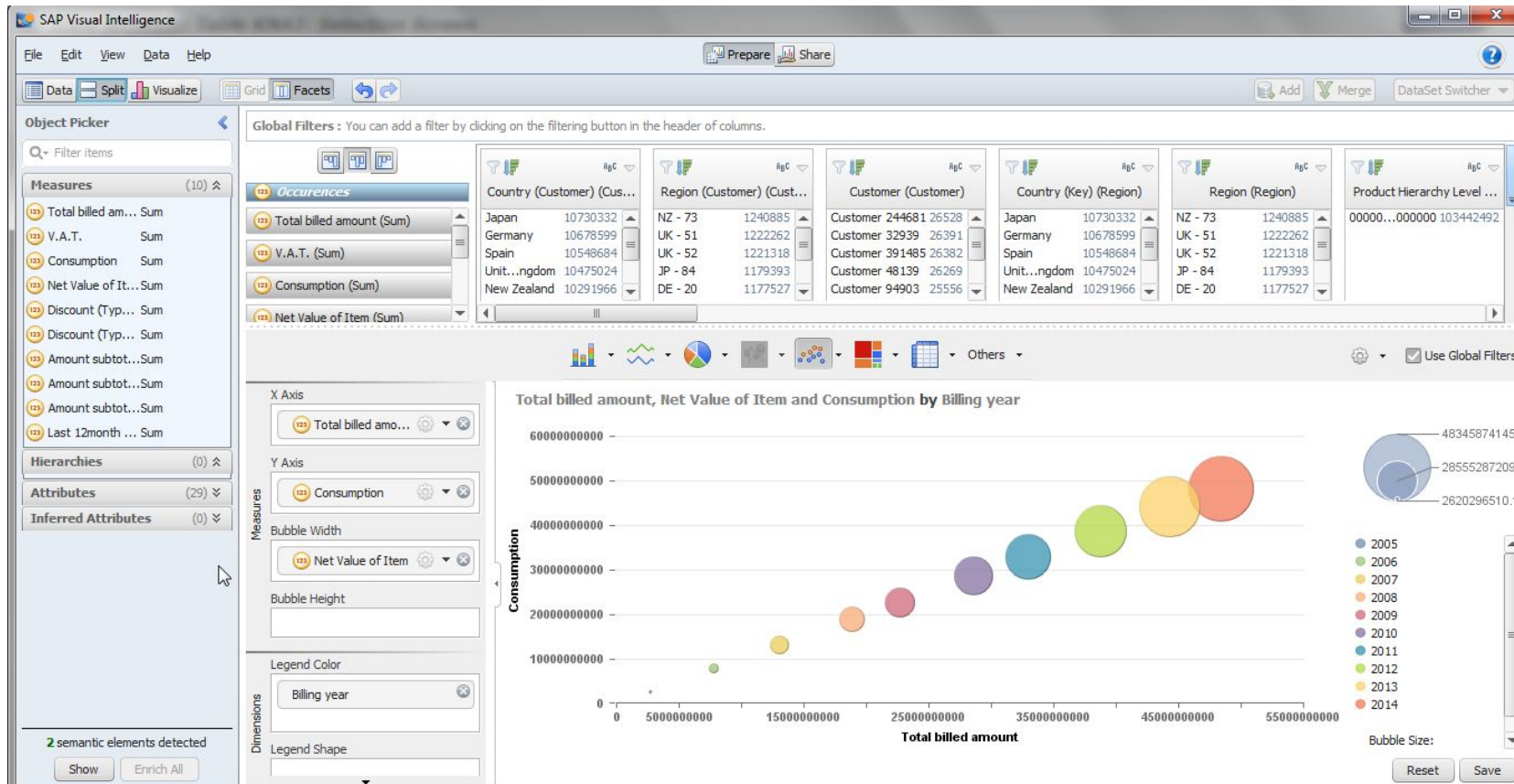


# Starting Points

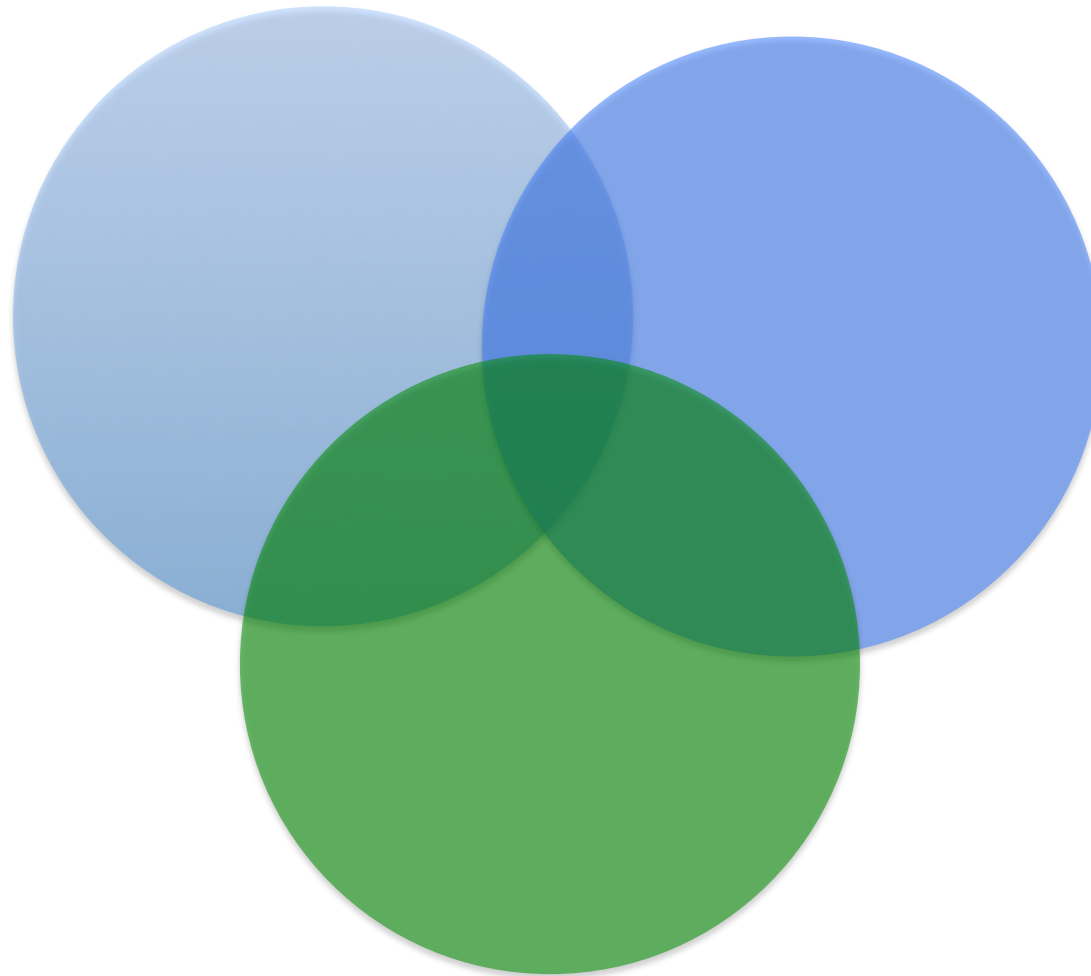
Postgres Explain: <http://www.sql.org/sql-database/postgresql/manual/sql-explain.html>

MySQL Explain: <http://dev.mysql.com/doc/refman/5.1/de/explain.html>

# 6. Analytical User Interface for HYRISE (i)



# Analytical User Interface for HYRISE (ii)

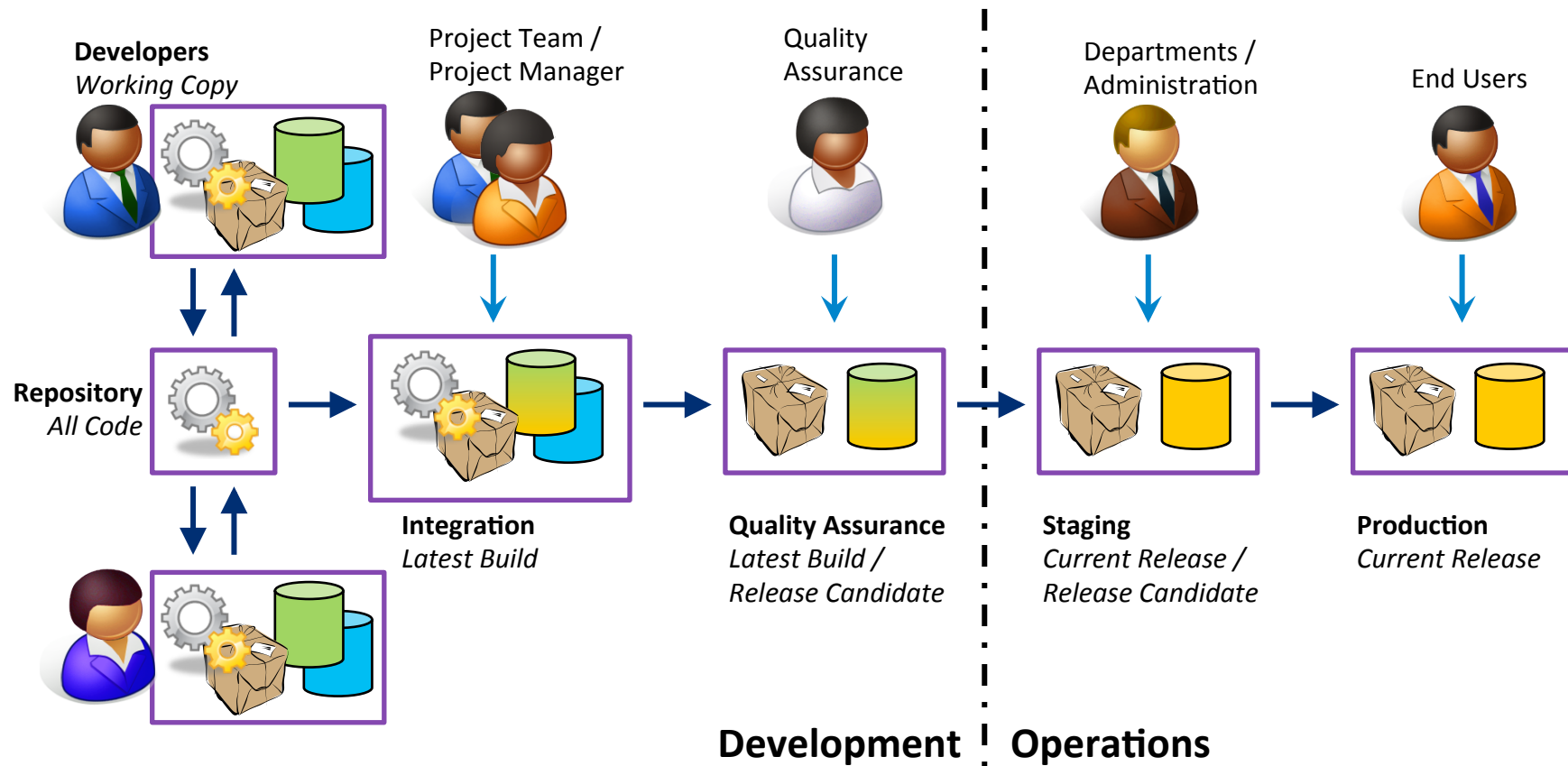


# 4. DevOps Lifecycle

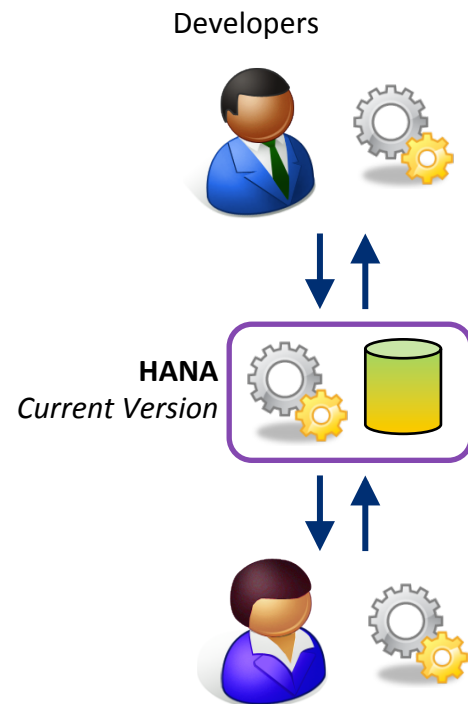
- DevOps: Development → Operations



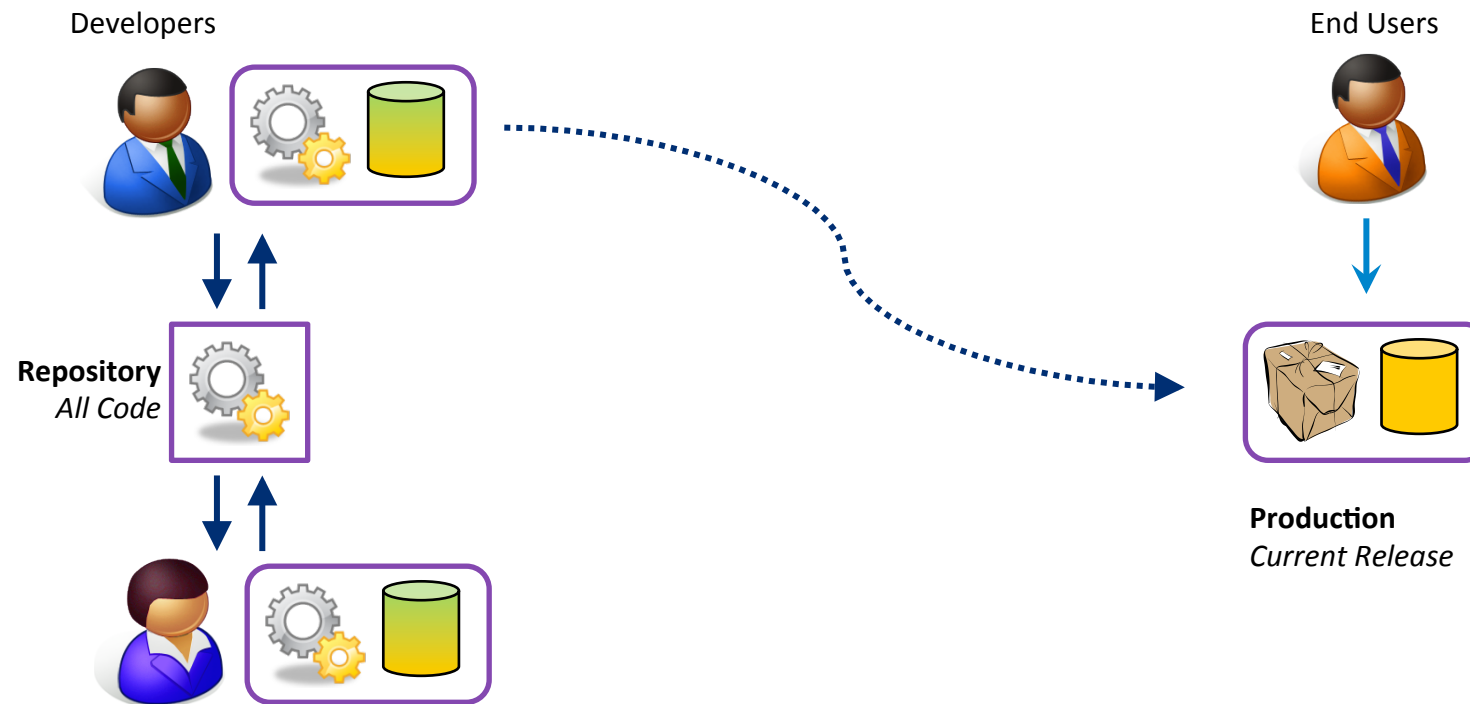
# DevOps Lifecycle



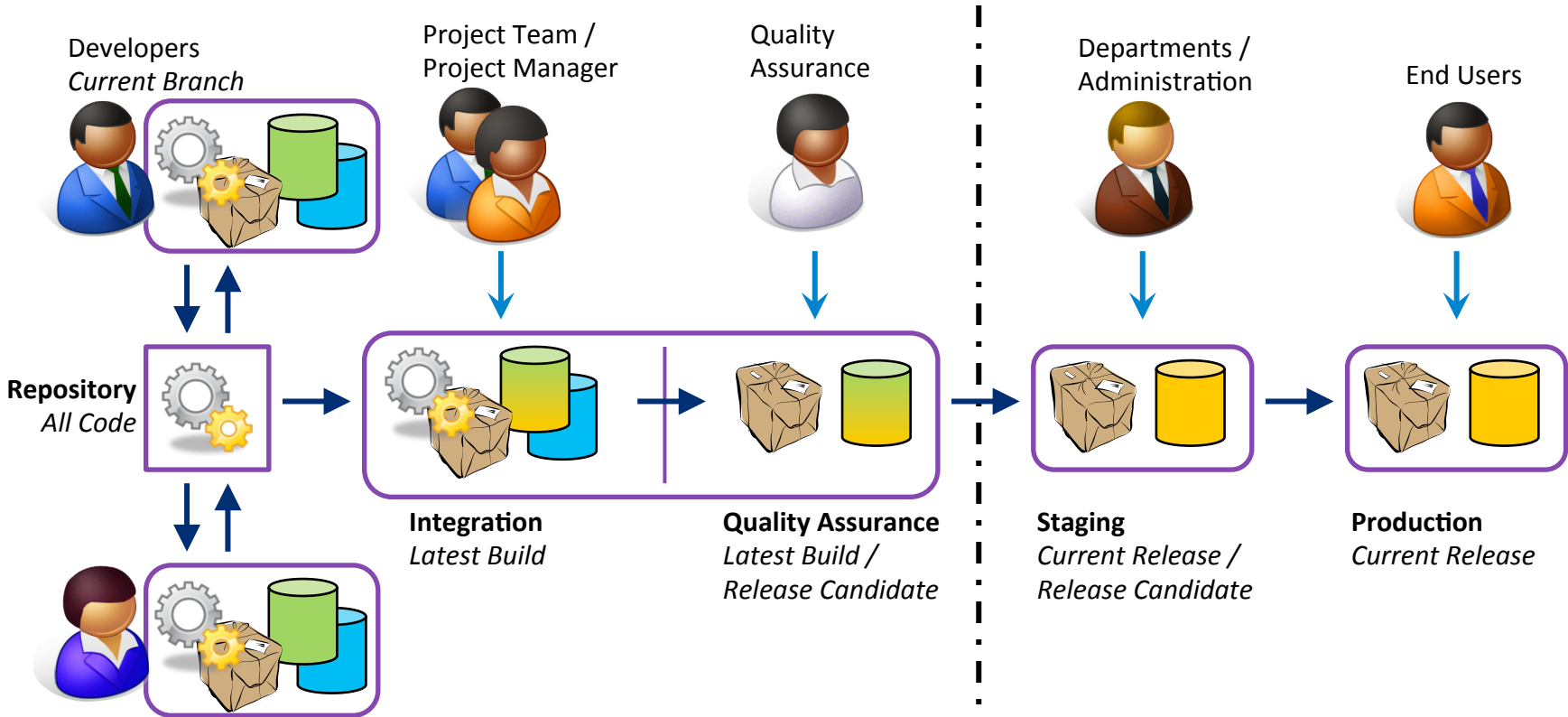
# Development on HANA






# Development on HANA



# DevOps on Hana



-  Development
-  Testing
-  Production

-  Code
-  Release

# XS Engine & BuildMaster

- BuildMaster is a DevOps Tool
- Task:
  - Set up release process for XS
  - (Programming will be required)



# 5. xsUnit

- XS has no automated testing framework
- Implement
  - Tests, test cases, test suites
  - Reports
  - (Fixtures)
  - (UI tests)
- Validate
  - With developers at ICP