Reinforcing Software Engineering Learning Through Provenance

Troy C. Kohwalter

Esteban W. G. Clua, Leonardo G. P. Murta

Instituto de Computação – Universidade Federal Fluminense (UFF)

Motivation

- Software Engineering Theoretical Classes
  - Knowledge

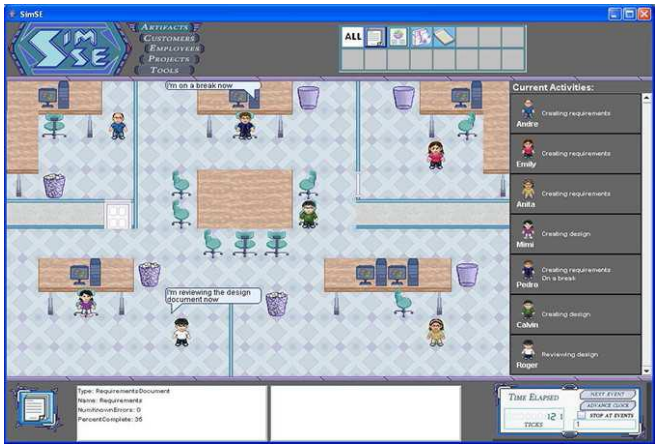
- Practical Work
  - Competence
    - Know how to
  - Performance
    - Show how to

Motivation

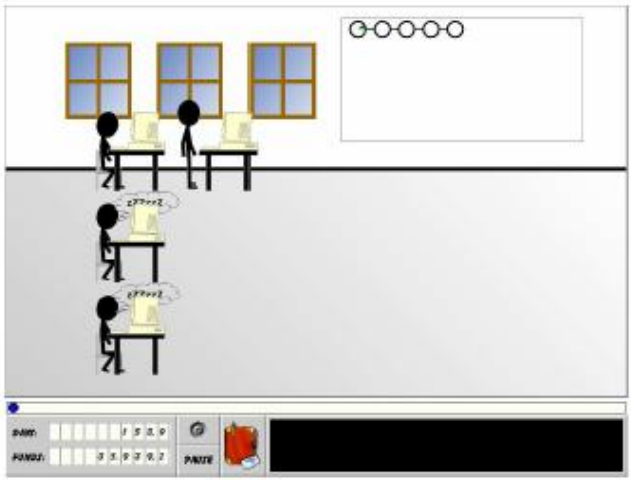
- Games
  - Fun
    - Enjoyment
  - Involving
    - Play
  - Motivating
    - Goals
  - Learning
    - Feedback

Serious Games
Software Engineering

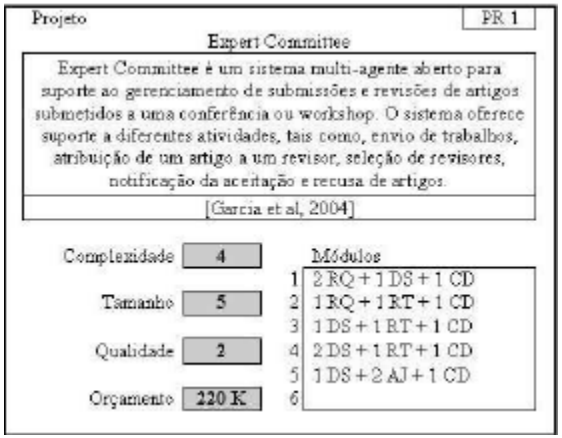
SIMSE



TIM

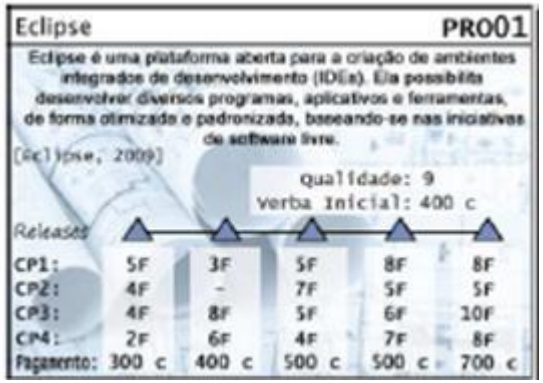


SIMULES



PnP

JEEES



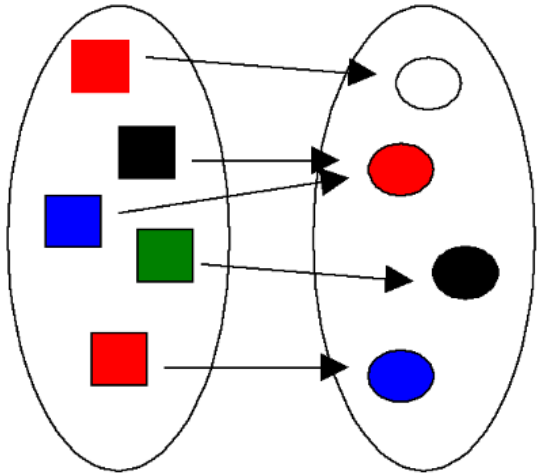
Motivation

- Bloody Hell! Why did this happen!?
  - How!
  - Why!?
  - Impossible!!!

- What have I done wrong?
- Did I make a mistake?
- How to analyze it?
  - Retry the game?
  - Watch a video?

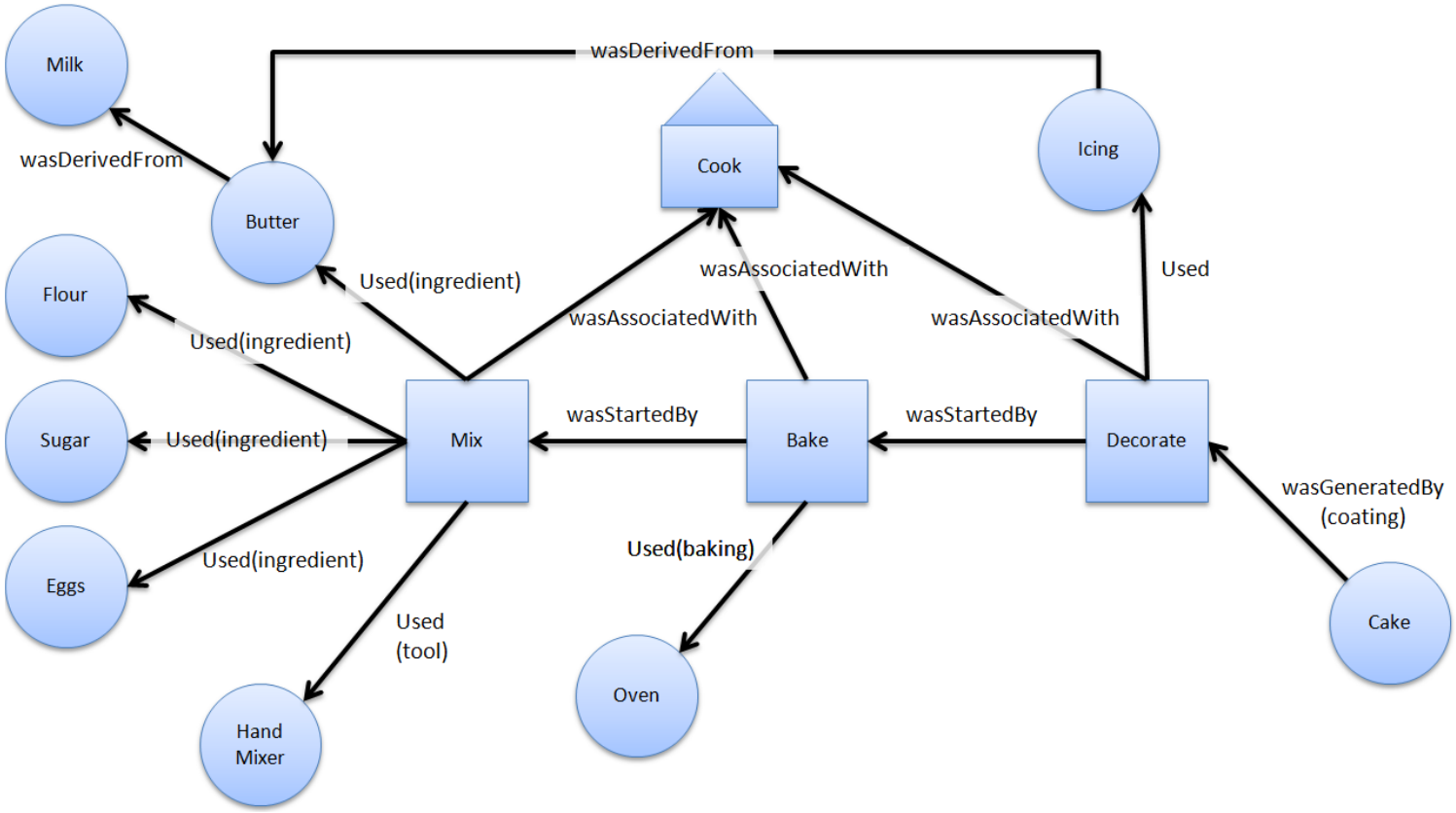
Previous Work

- Provenance in Games
  - Conceptual Framework
  - Map Domains
    - Provenance to Games
  - Gather
    - Provenance Information
    - Causal Relationships



Provenance

"Refers to the documented history of an art object, or the documentation of processes in a digital object's life cycle"



Provenance Gathering

- Entity
  - Objects
- Activity
  - Actions
  - Events
- Agent
  - NPCs
  - Player

Provenance Gathering

- Entity
  - Objects
- Activity
  - Actions
  - Events
- Agent
  - NPCs
  - Player

Provenance Gathering

- Entity
  - Objects
- Activity
  - Actions
  - Events
- Agent
  - NPCs
  - Player



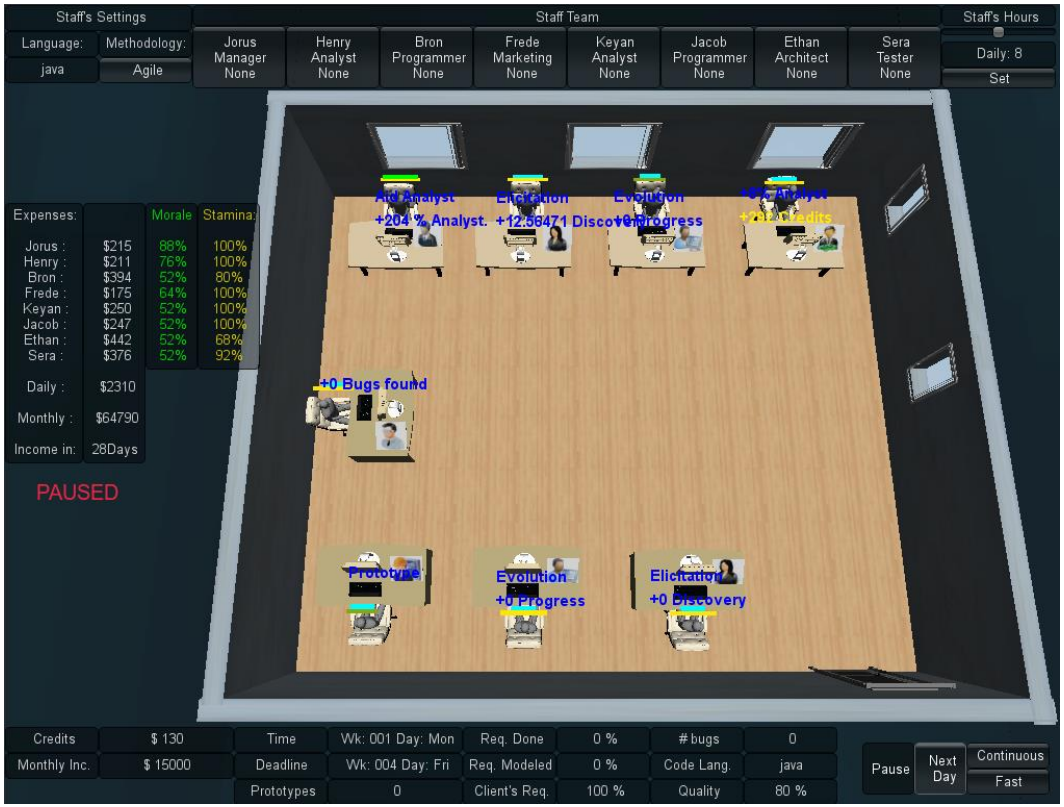



Goal

- Evaluate Game Provenance
  - Software Engineering
    - Aid students
    - Improve serious games teaching
    - Understand underlying reasons
  - Game session analysis
    - More efficient?
    - More effective?

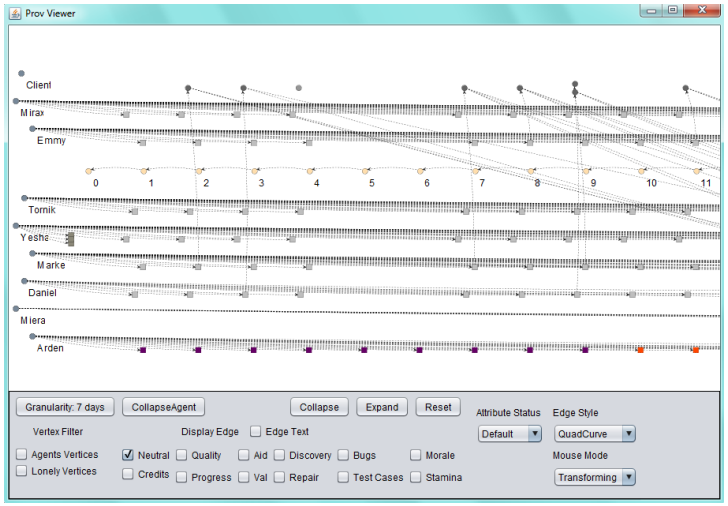
Case of Study

SDM: A Software Engineering Serious Game

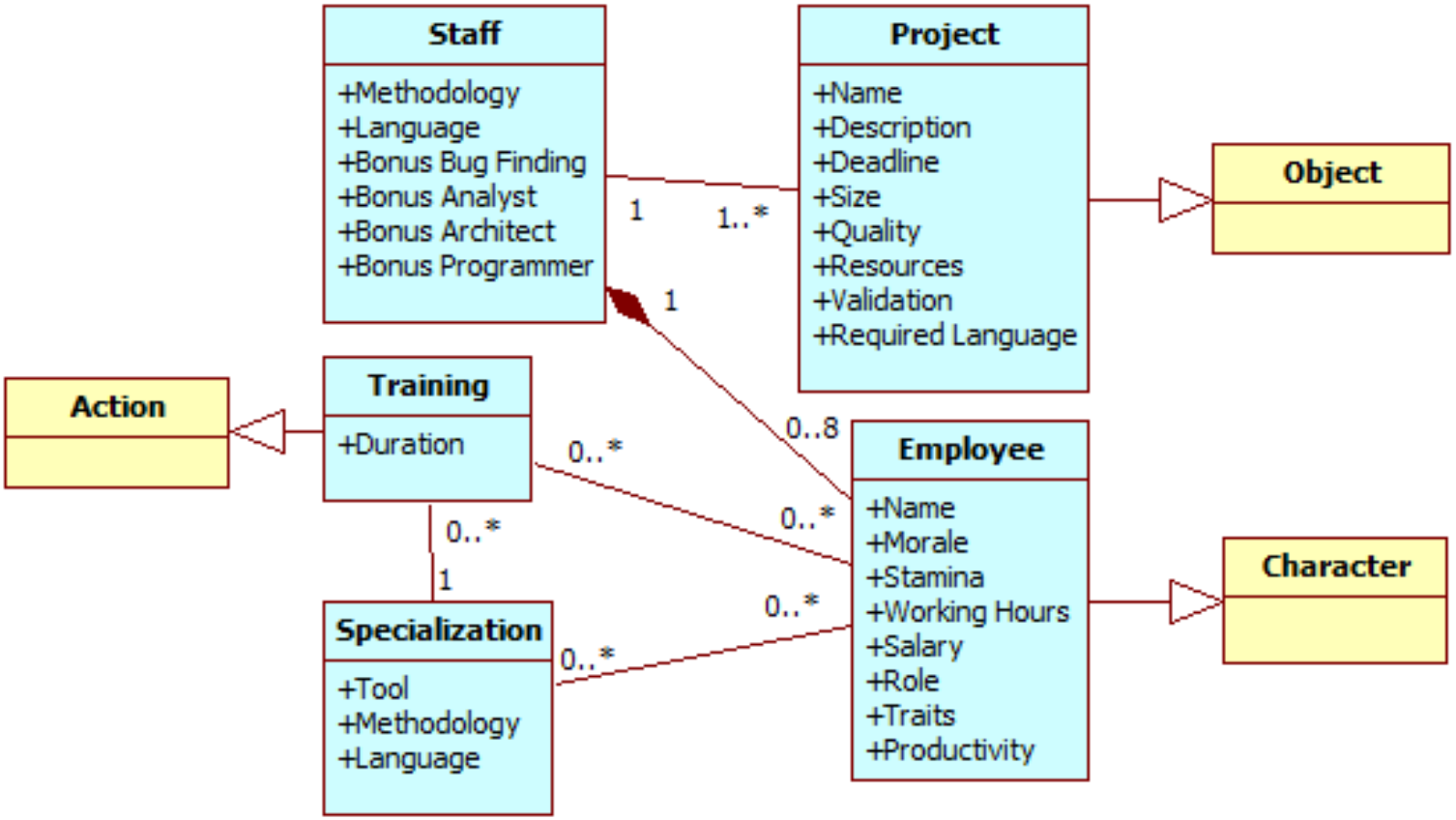


Provenance Visualization

Prov Viewer



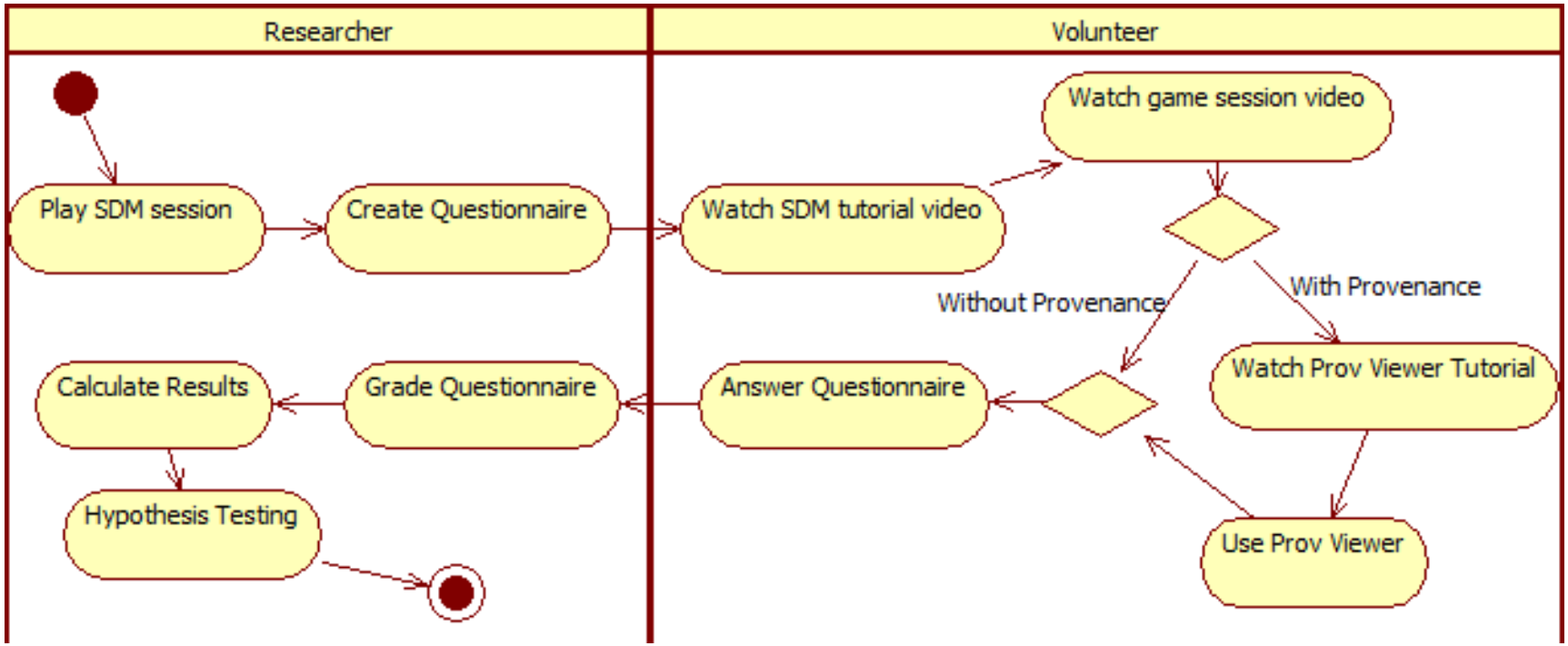
Case of Study



Evaluation

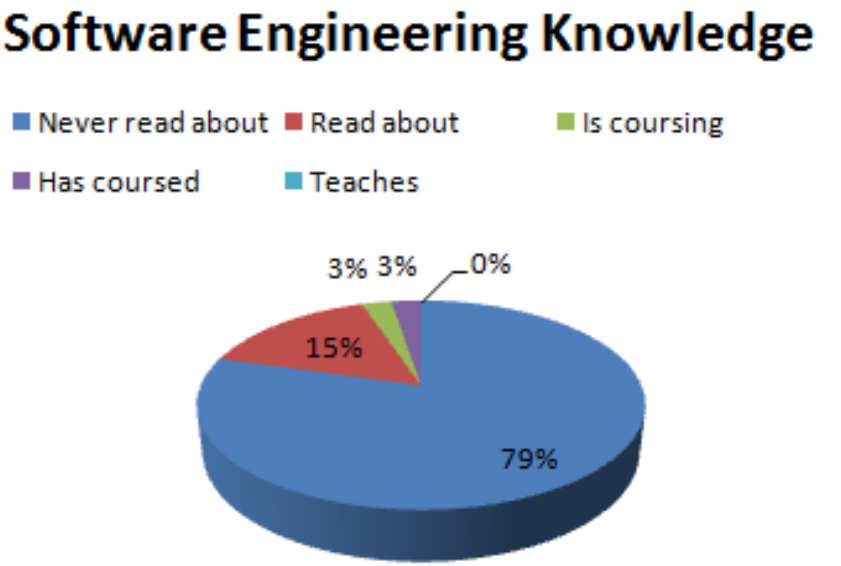
- Research Questions:
  - Provenance Analysis
    1. Is faster than only watching a replay of the game session?
    2. Is more accurate than only watching a replay of the game session?

Experiment Plan



Experiment

- 32 Volunteers



Questionnaire

- Question 1 : Starting Time
- Question 2: Experiment Group
- Question 3: Reason for employee Arden quit his job
  - Lack of payment
- Question 4: One reason for employee Daniel to quit his job
  - Lack of payment or Overworking
- Question 5: Why employee Tornik made no progress during a period of time
  - Lack of prototypes to validate requirements
- Question 6: Why Daniel's productivity had a sudden drop
  - Negative influence from his manager
- Question 7: Most contributing factor that allowed the software to finish in time
  - Negotiation for extended deadline
- Question 8: The two most contributing factors that caused financial problems
  - Hiring and staff training
- Question 9: Which employee was idle during a certain period of time
  - Arden
- Question 10: End Time

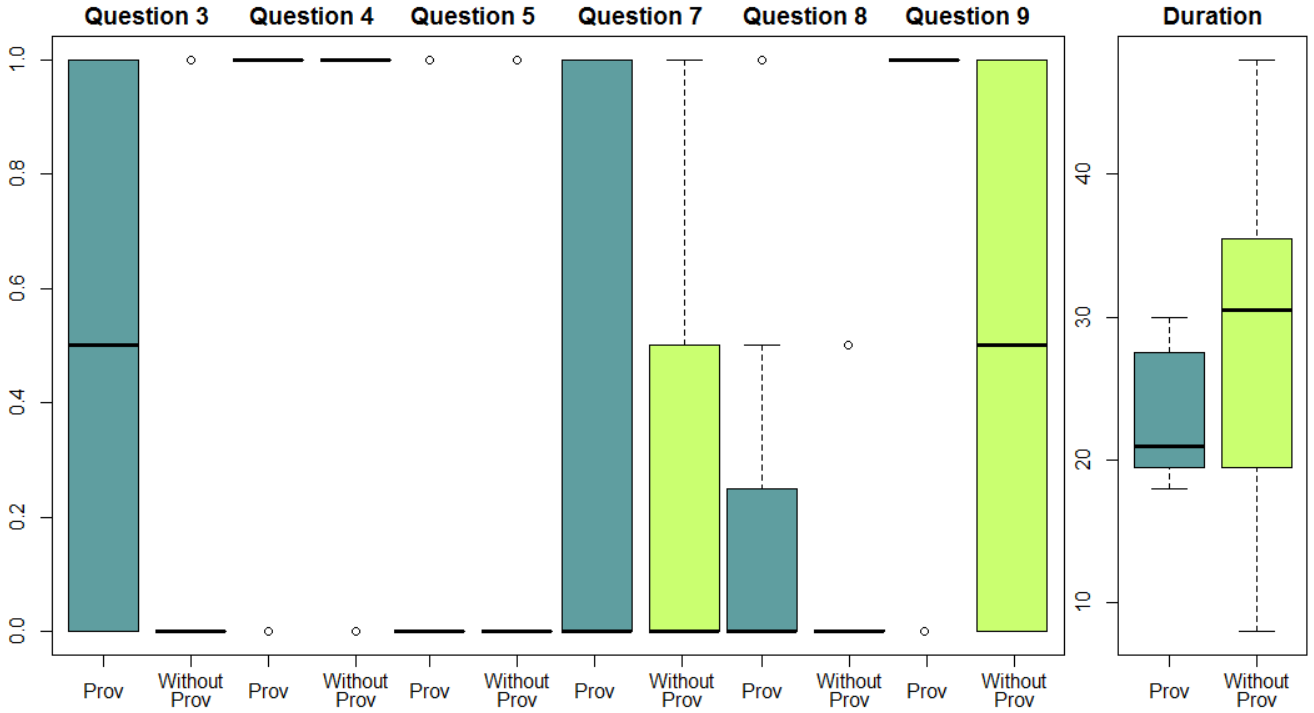
Experiment Results

- Grading each question
  - 0: Wrong answer
  - 0.5: Partially correct answer
    - Question 8 only
    - Offered only one reason
  - 1.0: Correct answer

- Mean for each question
  - Average of answers

Experiment Results

		Q3	Q4	Q5	Q6	Q7	Q8	Q9	Duration
With Prov	**Mean**	**0.5**	**0.9375**	0.1875	0	**0.375**	**0.1562**	**0.8125**	**23.1875**
	Standard Deviation	0.5164	0.25	0.4031	0	0.5	0.3010	0.4031	4.2461
Without Prov	**Mean**	0.0625	0.875	0.1875	0	0.25	0.0938	0.5	28.9375
	Standard Deviation	0.25	0.3416	0.4031	0	0.4472	0.2015	0.5162	10.5797

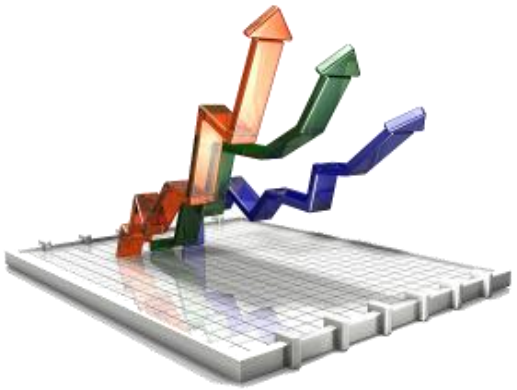


Experiment Results

- Mann-Whitney Test

$H_0 : \mu_{prov} = \mu_{replay}$

$H_1 : \mu_{prov} \neq \mu_{replay}$



p-value < α: Reject null Hypothesis

α = 0.05 = 5%

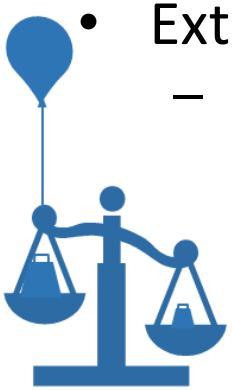
α = 0.05	Correctness							Time
	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Duration
p-value	**0.007259**	0.5757	1	Null	0.467	0.6371	0.07049	**0.03595**

Result Analysis

- Null hypothesis **Rejected**
  - Question 3
  - Questionnaire Duration
  - Values from both groups are not statistically equals
  - Provenance Group had better answers and faster
- Null hypothesis **almost Rejected**
  - Question 9 (0.07 ~ 0.05)
  - Provenance Group had better results but were not statistically significant
- Null hypothesis **not Rejected**
  - Other Questions
  - Cannot conclude if results from both groups are equals or not

Threats to Validity

- Internal
  - Group Division
    - Random
  - Individual Perception
  - Volunteers are more motivated
  - First Contact with Game and Tool
    - Mitigated by tutorials
- External
  - Discrepancy in Experience
    - Mitigated by being in the same semester and class



- Construct
  - Lack of Knowledge
    - Mitigated by having 7 Questions exploring different aspects
  - Duration
    - Mitigated by controlling start and end time
- Conclusion
  - Interpretation
    - Mitigated by explaining Questionnaire
  - Watched a video instead of playing

Conclusion

- Presented new perspectives on Software Engineering learning process
  - Game provenance
    - Induce deeper analysis
    - Induce discussions regarding the game session
    - Identify Cause-and-Effect Relationships

- Game Provenance can also help on:

1. Confirming the hypotheses formulated by students
2. Supporting tutors for a better guidance
3. Motivating practical exercises around some case studies
4. Extracting behavior patterns from individual sessions or groups of sessions

Research Questions

- Provenance Analysis

1. Is faster than only watching a replay of the game session?
   - **Yes**, even when using the tool for the first time
     - Statistically significant difference

2. Is more accurate than only watching a replay of the game session?
   - **Possibly**. Statistically provided more accurate answers in only one case
     - Other cases were inconclusive
     - Requires more experiments

Future Work

- Graph
  - Layouts
  - Inference

- Game module for Provenance Gathering
  - Easier integration with other games

- More Experiments!

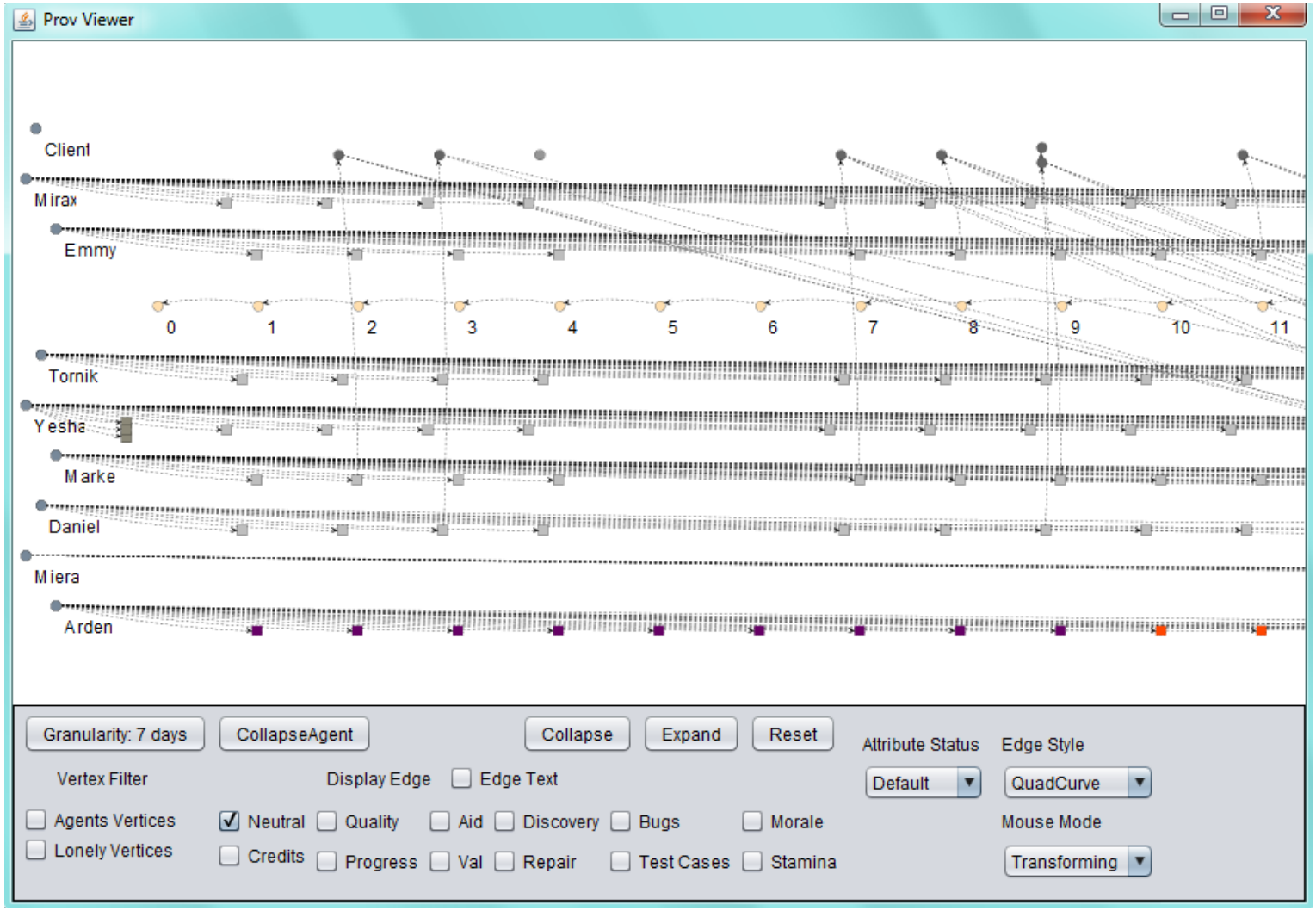
Reinforcing Software Engineering Learning Through Provenance

Troy C. Kohwalter

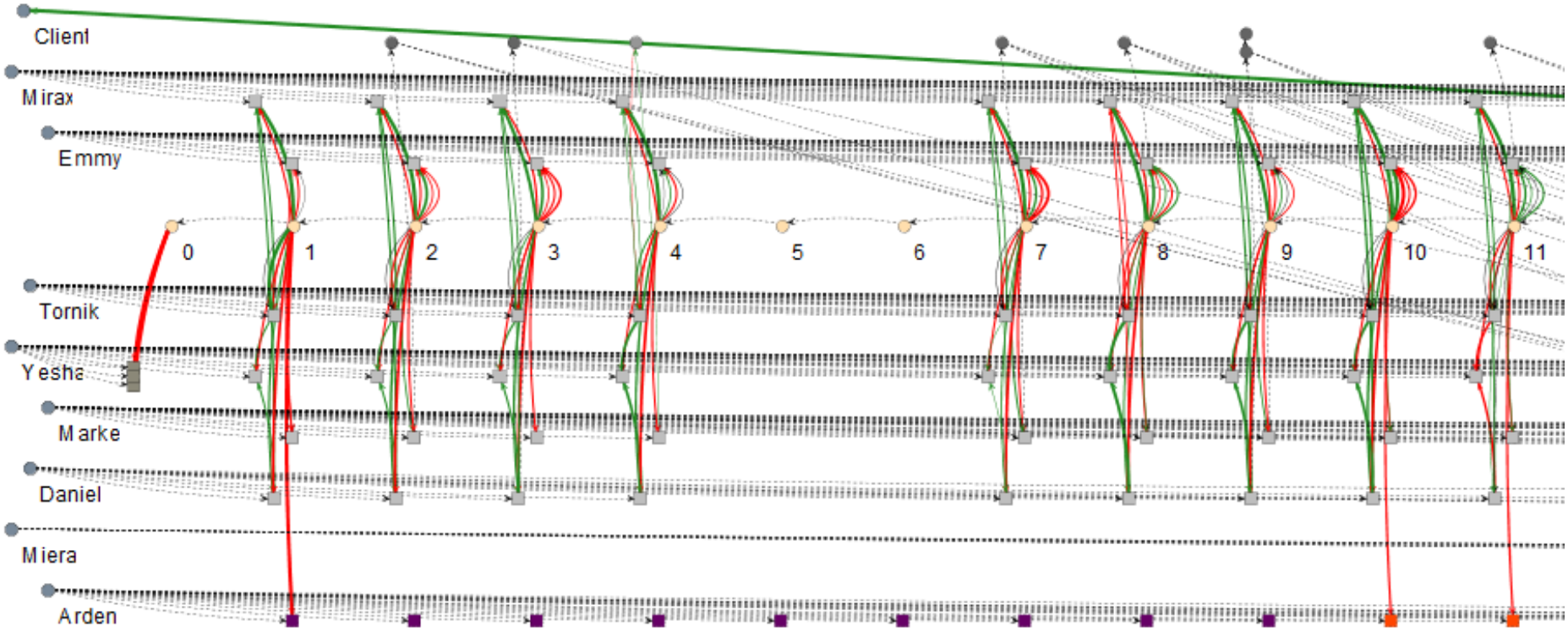
Esteban W. G. Clua, Leonardo G. P. Murta

Instituto de Computação – Universidade Federal Fluminense (UFF)

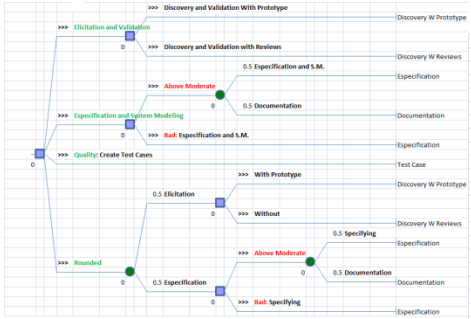
Visualization Example

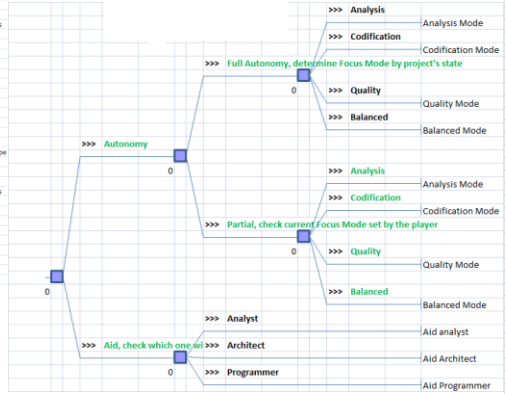


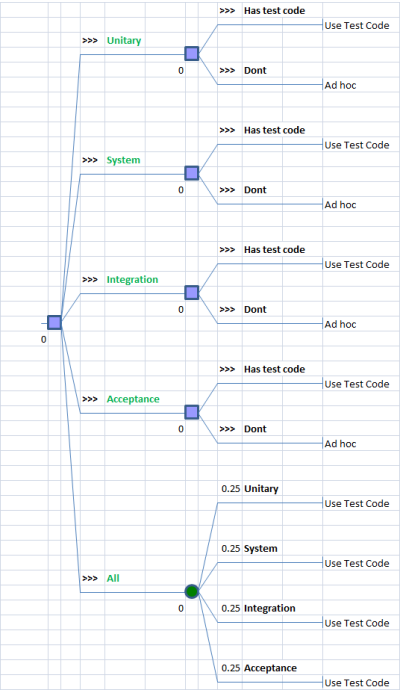
Visualization Example



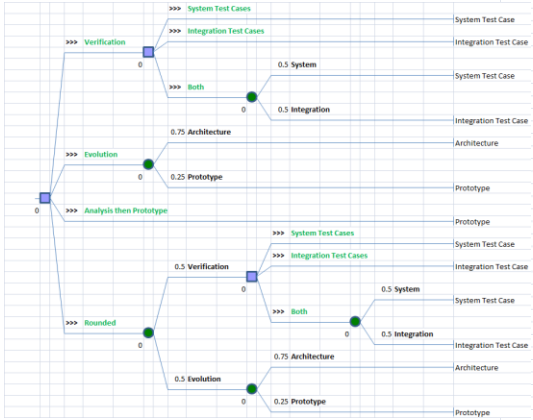
Case Study

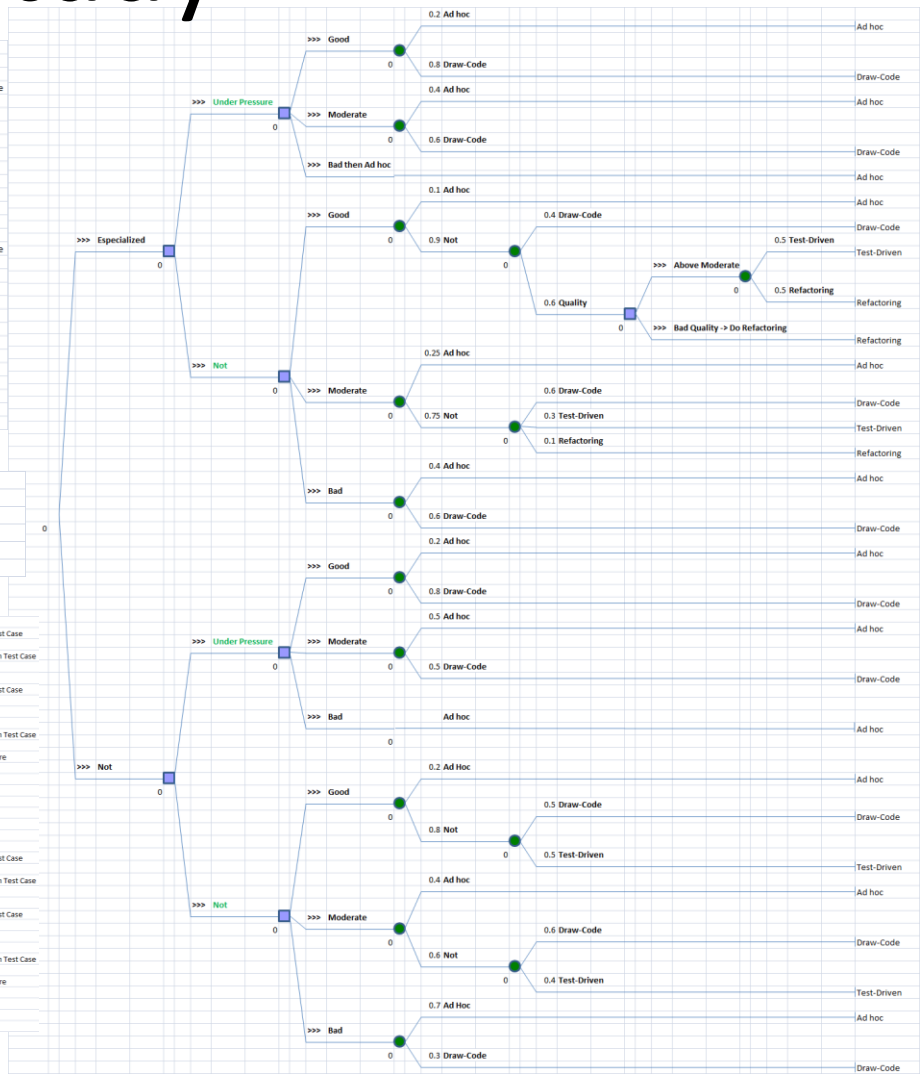

>>> Discovery and Validation With Prototype
Discovery W Prototype
>>> Elicitation and Validation
>>> Discovery and Validation with Reviews
Discovery W Reviews
0.5 Especification and S.M.
Especification
>>> Above Moderate
0.5 Documentation
Documentation
>>> Especification and System Modeling
>>> Bad: Especification and S.M.
Especification
>>> Quality: Create Test Cases
Test Case
>>> With Prototype
Discovery W Prototype
0.5 Elicitation
>>> Without
Discovery W Reviews
0.5 Specifying
Especification
>>> Rounded
>>> Above Moderate
0.5 Documentation
Documentation
0.5 Especification
>>> Bad: Specifying
Especification


>>> Analysis
Analysis Mode
>>> Codification
Codification Mode
>>> Full Autonomy, determine Focus Mode by project's state
>>> Quality
Quality Mode
>>> Balanced
Balanced Mode
>>> Autonomy
>>> Analysis
Analysis Mode
>>> Codification
Codification Mode
>>> Partial, check current Focus Mode set by the player
>>> Quality
Quality Mode
>>> Balanced
Balanced Mode
>>> Analyst
Aid analyst
>>> Aid, check which one is
>>> Architect
Aid Architect
>>> Programmer
Aid Programmer


>>> Has test code
Use Test Code
>>> Unitary
>>> Dont
Ad hoc
>>> Has test code
Use Test Code
>>> System
>>> Dont
Ad hoc
>>> Has test code
Use Test Code
>>> Integration
>>> Dont
Ad hoc
>>> Has test code
Use Test Code
>>> Acceptance
>>> Dont
Ad hoc
0.25 Unitary
Use Test Code
0.25 System
Use Test Code
>>> All
0.25 Integration
Use Test Code
0.25 Acceptance
Use Test Code


>>> Envolve
Envolve
>>> Repair
Repair


>>> System Test Cases
System Test Case
>>> Integration Test Cases
Integration Test Case
>>> Verification
0.5 System
>>> Both
System Test Case
0.5 Integration
Integration Test Case
0.75 Architecture
Architecture
>>> Evolution
0.25 Prototype
Prototype
>>> Analysis then Prototype
Prototype
>>> System Test Cases
System Test Case
>>> Integration Test Cases
Integration Test Case
0.5 Verification
0.5 System
>>> Both
System Test Case
0.5 Integration
Integration Test Case
>>> Rounded
0.75 Architecture
Architecture
0.5 Evolution
0.25 Prototype
Prototype


0.2 Ad hoc
Ad hoc
>>> Good
0.8 Draw-Code
Draw-Code
>>> Under Pressure
0.4 Ad hoc
Ad hoc
>>> Moderate
0.6 Draw-Code
Draw-Code
>>> Bad then Ad hoc
Ad hoc
>>> Especialized
0.1 Ad hoc
Ad hoc
>>> Good
0.4 Draw-Code
Draw-Code
0.9 Not
0.5 Test-Driven
Test-Driven
>>> Above Moderate
0.5 Refactoring
Refactoring
0.6 Quality
>>> Bad Quality -> Do Refactoring
Refactoring
0.25 Ad hoc
Ad hoc
>>> Not
>>> Moderate
0.6 Draw-Code
Draw-Code
0.75 Not
0.3 Test-Driven
Test-Driven
0.1 Refactoring
Refactoring
0.4 Ad hoc
Ad hoc
>>> Bad
0.6 Draw-Code
Draw-Code
0.2 Ad hoc
Ad hoc
>>> Good
0.8 Draw-Code
Draw-Code
0.5 Ad hoc
Ad hoc
>>> Under Pressure
>>> Moderate
0.5 Draw-Code
Draw-Code
>>> Bad
Ad hoc
Ad hoc
>>> Not
0.2 Ad Hoc
Ad hoc
>>> Good
0.5 Draw-Code
Draw-Code
0.8 Not
0.5 Test-Driven
Test-Driven
0.4 Ad hoc
Ad hoc
>>> Not
>>> Moderate
0.6 Draw-Code
Draw-Code
0.6 Not
0.4 Test-Driven
Test-Driven
0.7 Ad Hoc
Ad hoc
>>> Bad
0.3 Draw-Code
Draw-Code

Provenance Visualization