In-Time Role-Specific Notification as Formal Means to Balance Agile Practices in Global Software Development Settings

Dindin Wahyudin, Benedikt Eckhard, Matthias Heindl, Alexander Schatten, Stefan Biffl

Institute of Software Technology and Interactive Systems

Vienna University of Technology

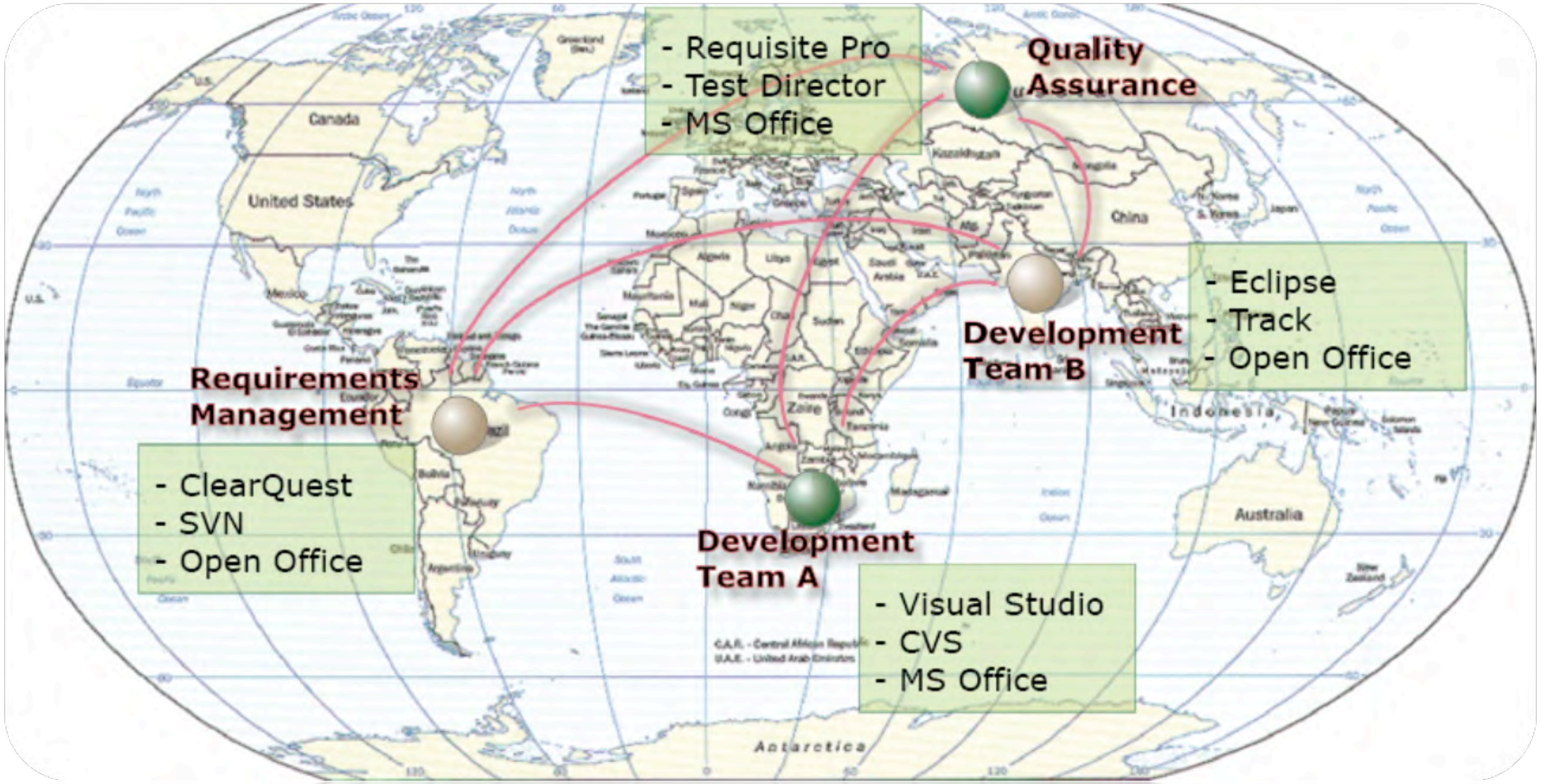
dindin@ifs.tuwien.ac.at

Motivation

- Over the last few years, many organizations began to experiment with remotely located software development facilities and with outsourcing to seek lower costs and access to skilled resources.
- Dimension of problem of Current GSD Practices
  - [Herbsleb and Moistra 2001]
    - Geographically distributed project participants
    - Strategic and cultural Issue
    - Inadequate communication, which may lead to lack of awareness and high cost coordination
  - [deSouza2002]
    - Distributed developments such as in GSD face critical issues such as complex dependency among tasks and artefacts
  - [Perry 1994, Vessey 1995]
    - in GSD projects typically the team members spend more than 50% development time for communication, and about 70% of this time accounted for cooperative activities.
  - [Passivaara 2006]
    - Unstable project environment due to requirement changes from the customer side which depict the need for agile practices adoption in GSD setting

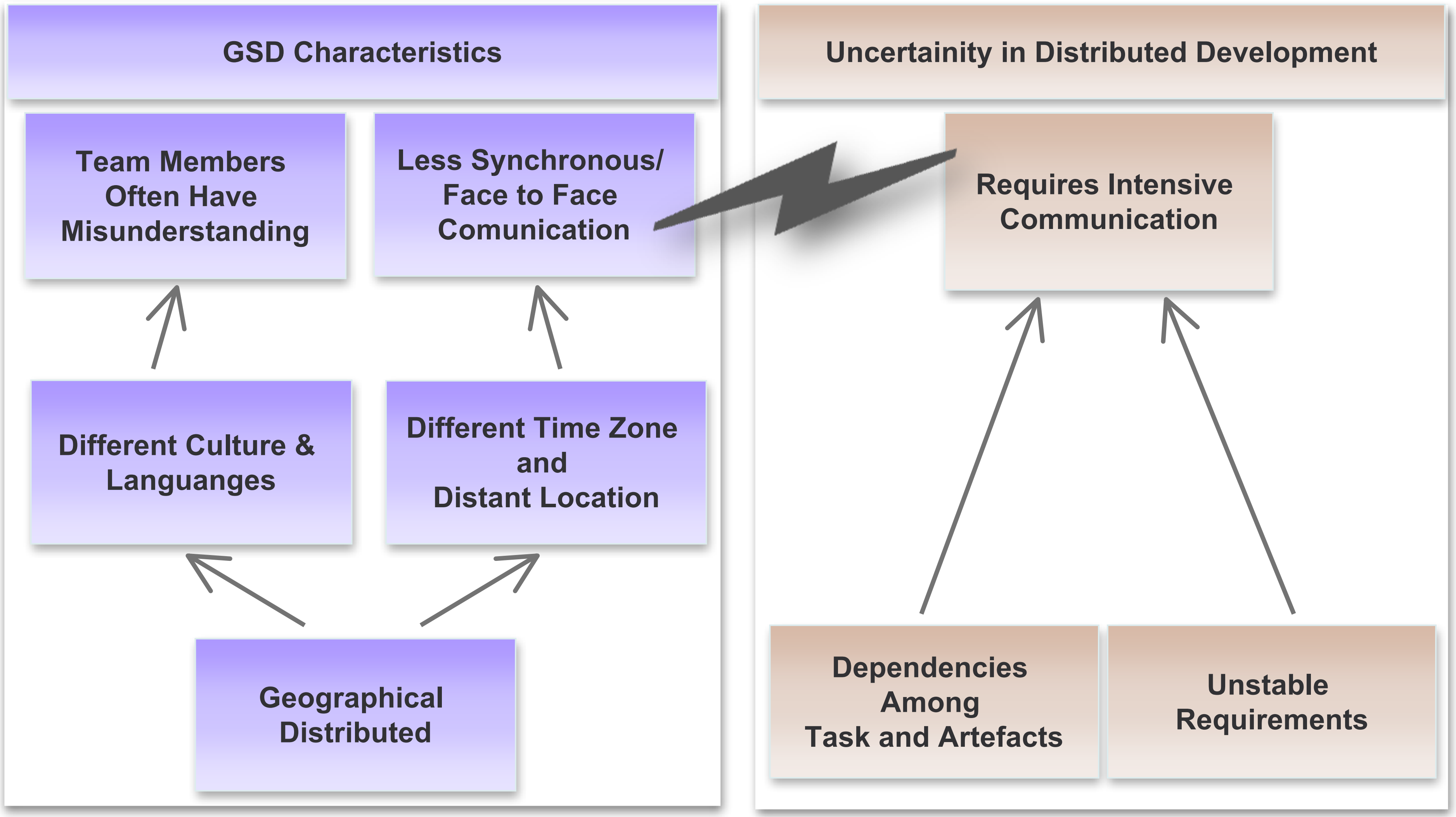
Global Software Development

GSD Participants are distributed around the Globe



These distributed teams collaborate to deliver high-quality software.

Conflict within Agile-GSD Setting


GSD Characteristics
Team Members Often Have Misunderstanding
Less Synchronous/ Face to Face Comunication
Different Culture & Languanges
Different Time Zone and Distant Location
Geographical Distributed
Uncertainity in Distributed Development
Requires Intensive Communication
Dependencies Among Task and Artefacts
Unstable Requirements

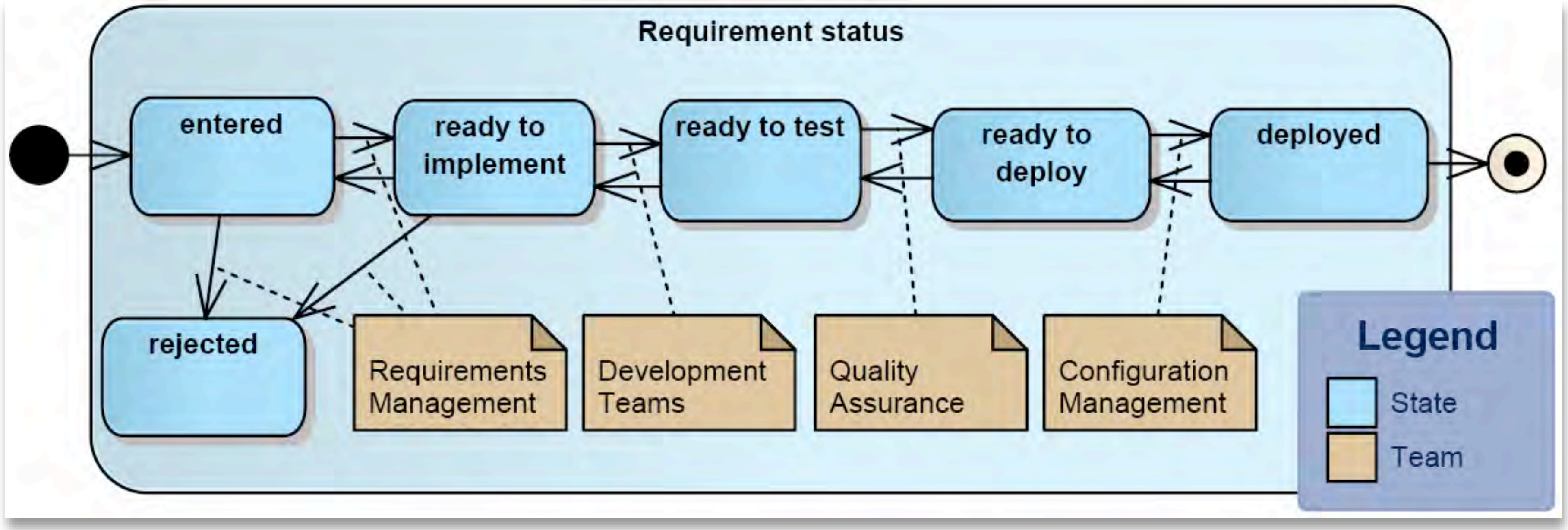
A major challenge is to keep all team members aware of recent changes of requirements and project status without providing too little (due to ad-hoc communication) or too much information (tool subscription) for each role.

The Need of Formal Notification

- In agile-distributed setting ad-hoc communication may not be sufficient, and notification can help to counter this effect
- a notification is an object that collects together information about change state, errors, early warning and other time-relevant project status information and communicates it to the presentation of target user;
- Formalism of key communication allow automation to reduce effort and higher correctness and completeness of conveyed information

Key Communications	Roles Involved	Occurrence	Possibility of Loss and Delay	Need for Automation
Failed Test Cases	Tester, Project Manager, Developer	High	High	Yes
Requirement Traces	Project manager/ Requirement management, developer, tester	High	High	Yes
Fix mistakes in code set	Developer	High	Low	No

State Changes during Project Execution
Case: Requirement Tracing (RT)

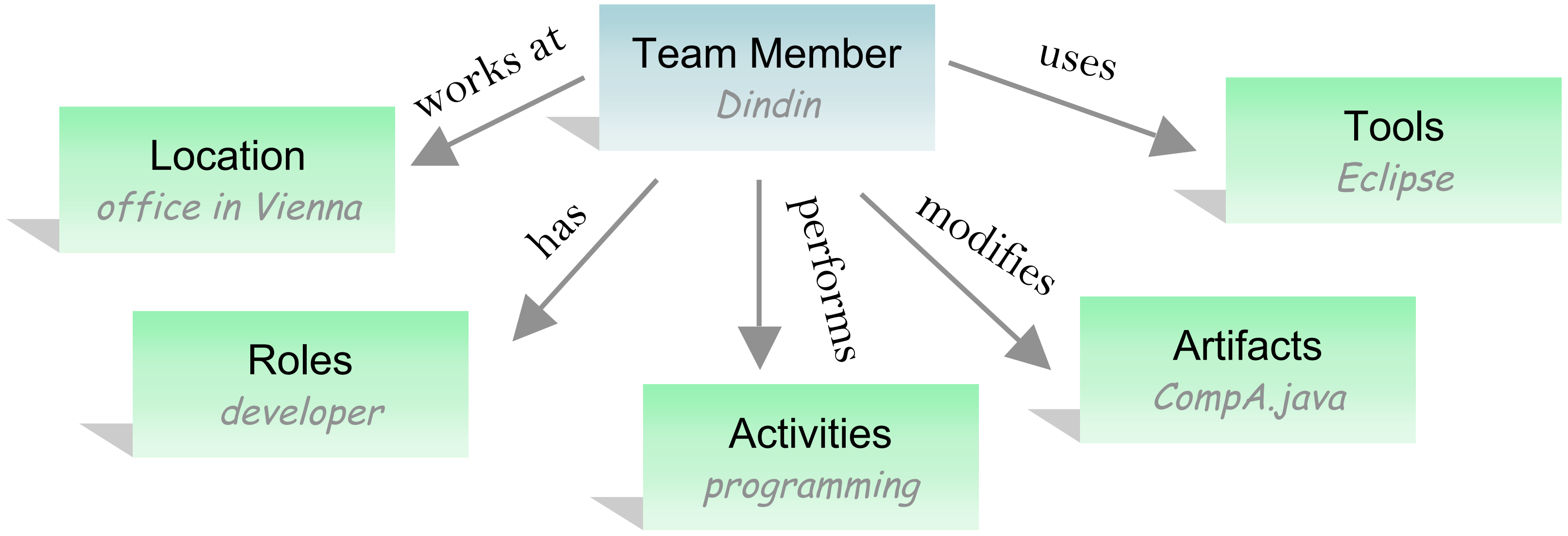


Team Members need information of project state changes regarding to their work context in timely manner, and intend to avoid unecessary information

What Is Context Specific Information

Issues during collaboration:

-High Dependencies among Tasks and Artifacts
-Unstable requirement
-Changes of work context and availibility issue



To support their role, a team member needs:

the	right information
at the	right time
at the	right place

For particular changes concerning his current work context

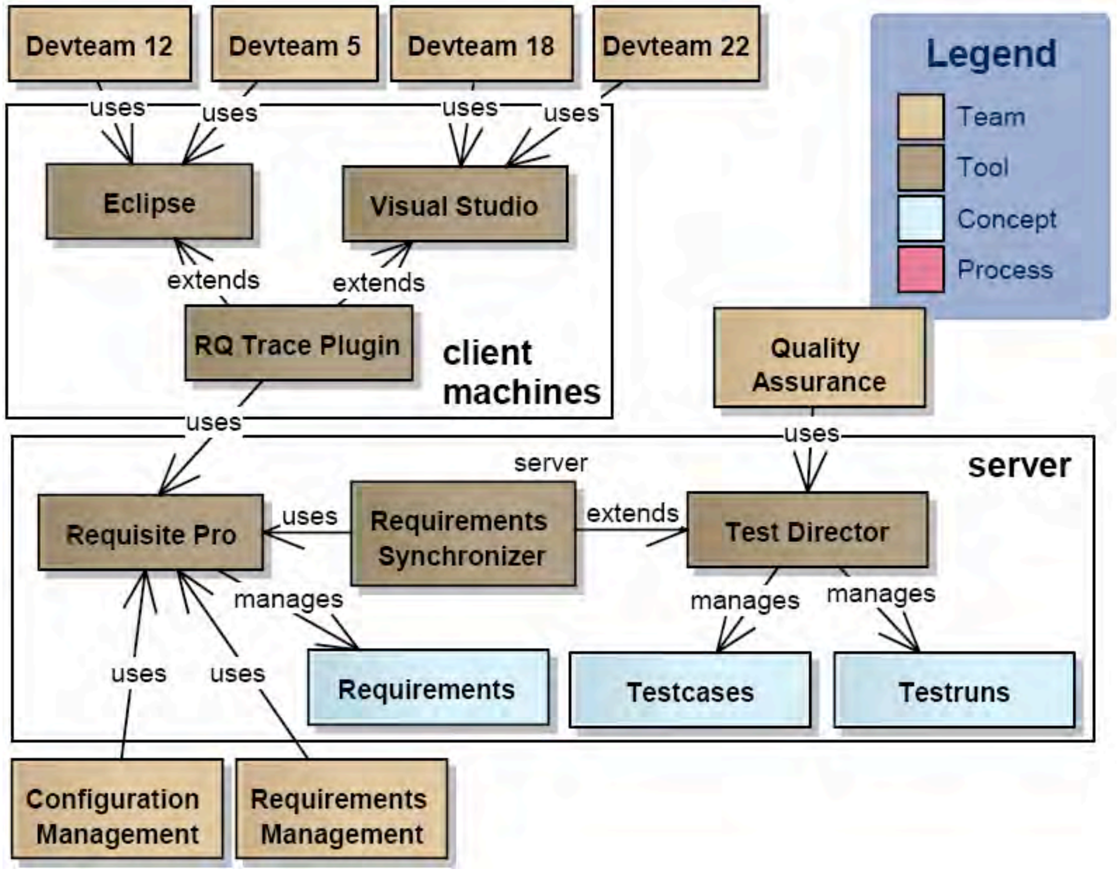
Dependecy Issue in Agile Technology Point of View

Traditional (Manual) approach (E.g. Key phrase dependencies, RT Matrices)

- High effort and limited manageable information and traces

Automation of RT (E.g. Req.Pro, Doors)

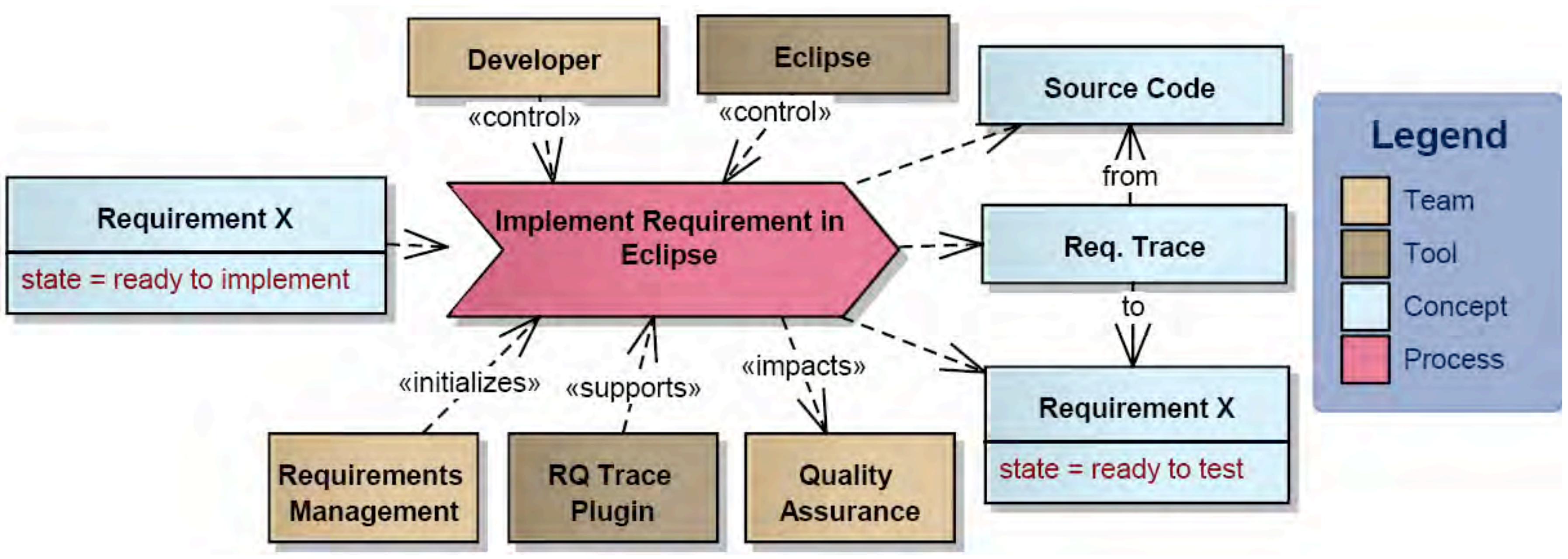
- Provide the facility to relate items stored in a database and
- automatically indicate which artifacts are effected due to requirement changes
- the tools do not automate the generation of trace links, which remains a manual,



Proposed solution:

- A tracing approach that uses a plug-in to automatically import the requirements from a requirement management tool into developer tools (integrated tool support)

Process Point of View



- We need to identify the roles and process involved during communication execution
- We have to describe the relation of roles, process, tools and artifacts
- Later we identify which roles need to be notified of particular changes

Formalism

- A notification should be specified with Notification Specification Language (NSL)
- **Notification Specification**
  - To specify the correct notification for target user in formal way
  - Elements of key communication that should be formalized:
    - **processes** performed during communication task
    - **roles** involved in information exchange
    - **data transmitted** during communication
    - distributed **events** to publish-subscribed the notification, and
    - **delay** allowance of notification represents the time between e.g. artifact/requirement changes and capturing of notification by target user.
- **Notification Rules**
  - The syntax to formulate notification rules consists of the following parts:

    if <*change event*> then notify <*whom*> in what way <*representation mode*> (e.g., e-mail, SMS, entry in change log) by <*when*> (e.g., immediately; batch every hour/day)

Examples of Notification Specification Case Failed Test Case

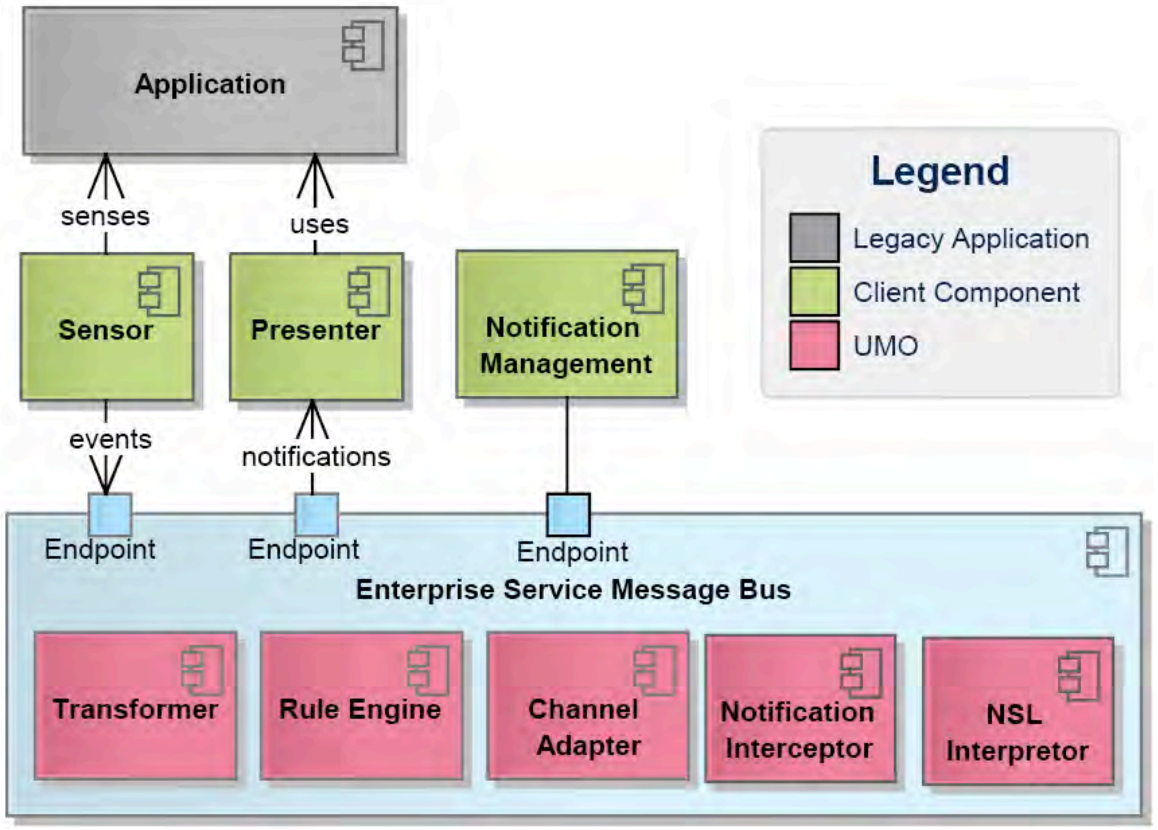
```
Title : " Notify developers of failed test cases ."
Receiver : any subscribed User ? user
Context :
the ? user " has " the Role " developer " and
the ? user " uses " a Requirement ? requirement and
a TestCase ? testcase " tests " the ? requirement and
a TestRun ? testrun " relates to" the ? testcase and
the status of the ? testrun changes from " undefined " to " failed "
Deliver : immediate
Channel : the activeChannel of the ? user
Subject : " Test case {? testcase .id} failed ."
Body :
"{? testrun . date }: The test of requirement {? requirement .id} failed
in the test run {? testrun .id}"
```

Examples of Notification Specification Case Requirement Tracing

```
Title : " Notify project manager of Change in Code Set related to specific Requirement."
Receiver : any subscribed User ? user
Context :
the ? user " has " the Role " project manager " and
the ? user " assign " a Requirement ? requirement and
a Requirement ? requirement " trace " the ? requirement and
a Trace ? trace " relates to" the ? codeset and
the status of the ? traces changes to “updated"
Deliver : in batch daily at 09:00 AM
Channel : the Requisite Pro of the ? user
Subject : " Code Set {? codeset .id} changed ."
Body :
"{? trace .date }: Code set {? codeset .id} relates to requirement {? requirement .id} has been changed
in the trace {? trace :id}"
```

Tool Support for In-Time Role-Specific Notification

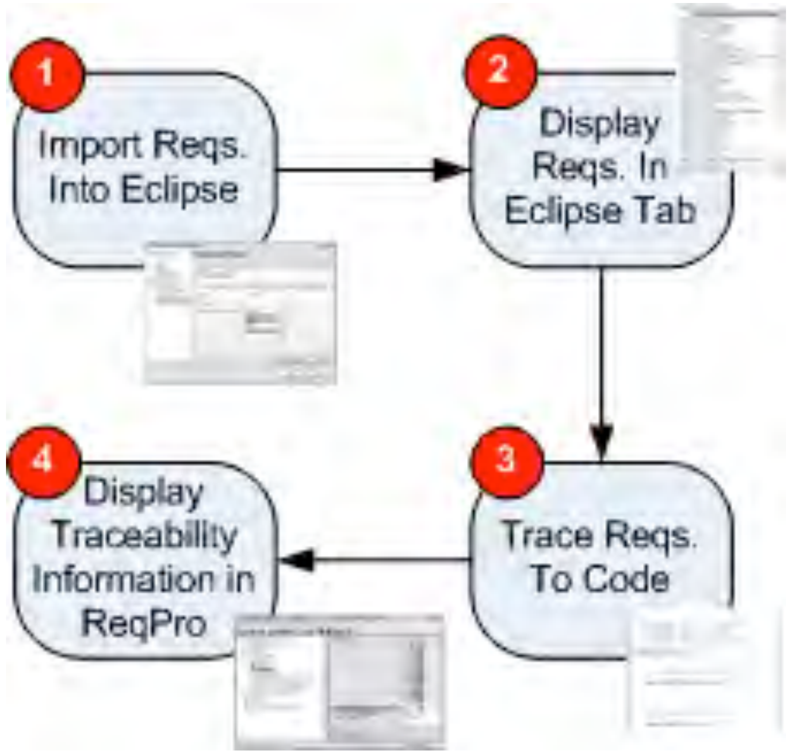
- Tool supported notification intended to replace some missing or high cost key communication in GSD collaboration in more cost effective means
- Should avoid “new tool syndrome” and low level of distraction
- Existing work tools (Legacy App.) are integrated using plug-ins interface
- Notification presented as part of target user work tools environment



Currently we have a prototype called **NOTICON** (using Enterprise Service Bus technology)

Initial Empirical Evaluation

- Goals:
  - To provide notification regarding requirement changes in common development work tools
  - To reduce effort and incorrectness to capture requirement trace (RT)
- Tool based Solution
  - Plug-in to automatically import the requirements from Requisite Pro into *Eclipse*
  - Requirements can be linked to the code element the developer currently works on
  - The traces are automatically re-imported into the Req.Pro, where Req. Management readily use the trace information for change impact analysis


1
Import Reqs. Into Eclipse
2
Display Reqs. In Eclipse Tab
3
Trace Reqs. To Code
4
Display Traceability Information in ReqPro

Case Study:

- A set of Medium size projects at Siemens
- 2 to 4 sites (e.g. Austria, Rumania, and Slovakia)
- 10 to 60 team members
- Compare our solution with traditional and systematic (tool supported) RT in context of change impact analysis

Results

- Requirement traces (RT) considered as important key communication with high risk impact based on risk analysis and practitioners interview at Siemens PSE
- *In-Time Role Specific Notification offers for RT context*
  - The average net effort to create a trace of integrated tools approach was only around 20% than the best practice in traditional tracing and current tool supported tracing;
  - Less cognitive complexity for developers to capture and maintain traces, because they receive notification within their familiar work environment without using extra tools.
  - More complete trace generation: By providing tracing facilities in their familiar work environment, the acceptance of developers to capture and maintain traces should increase.
  - More correct trace generation: due to less delay, the captured trace information should be more correct than with an approach where code is developed first, and traces are captured later on.
- Advance investment of notification in large complex project, with high occurrence of changes in requirement and code development seems paid-off

Conclusion

- To successfully conducting GSD project, one challenges is to keep all team member has enough information awareness regarding their work context
- As current ad-hoc communication is not sufficient in in Agile-Distributed setting
- Hence the formalization of key communications allow automation which provide benefits such as effort reduction, increase completeness and correctness of information being transmitted
- In-time and role-specific notification support GSD team members with more timely-effective information awareness of project status changes and reduce the possibility of rework and delay
- Our initial empirical evaluation reveals promising result, however the tool supported notification have limitation, as in small projects with less change in requirements or artifacts the investment can be higher than expected benefit
- Future work:
  - Advance development of Notification Specification Language (NSL) which easily transformed into code/implementation of notification
  - Empirical evaluation of the concept by employing more scenarios of communication with larger development team

Contact

For detailed information and Questions please contact

Dindin Wahyudin
Institute for Software Technology
and Interactive Systems
Vienna University of Technology
dindin@ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at/