TRACEABILITY
TAMING UNCONTROLLED CHANGE IN SOFTWARE DEVELOPMENT

Dawid Weiss
Poznań University of Technology
dawid.weiss@cs.put.poznan.pl
Presentation outline.

**The roadmap**

- Software development problems
  - Meet the Superproject
  - The eternal (well, quite) enemy: change
- Dodging the problem
  - Change management processes
  - Integrated tools
- Facing the problem: Traceability
  - Project meta data
  - Use of meta data
  - Any tool support? Yes, Ophelia!
  - Metadata overuse analysis
- Summary
Almost every software project goes through these phases of development. We have two axes here: vertical, which could be representing effort put into the project, number of people involved in it, or the usual time flow and horizontal, which could be, among others, representing complexity of management of such project.
People have very different point of view on the project and they tend to see different things.
Software development problems
(in large, distributed projects)

• Sneaking complexity

Beautiful and simple ideas tend to get complicated…
Communication difficulties may arise as a result of physical distance, different languages or…
… simply because people are not willing to talk to each other.
Software development problems
(in large, distributed projects)

- Vanishing responsibility

*DON'T BLAME ME! I DID MY PART!*
Software development problems
(in large, distributed projects)

- Short-sightedness syndrome

People are not aware of the role their work plays in the overall effect. They don’t realize which components of the system depend on their part.
The eternal enemy: CHANGE

- Sources of change
  - External
    - Volatile customer requirements
    - Bug reports
    - Unpredictable obstacles (hardware problems, proprietary interfaces, shrinking budget…)
  - Internal
    - Refactoring (if something works… don’t change it)
    - Change of development libraries, technology etc.
    - Project expansion (iterative development)

Change doesn’t have to be an enemy… but in most cases it is ©Refer to Kent Beck’s eXtreme Programming book
People’s frustration is the most severe of uncontrolled change’s results.
QUESTION

HOW TO SUPPORT ORGANIZATIONS IN AVOIDING PROBLEMS WITH CHANGE MANAGEMENT?
The roadmap

- Software development problems
  - Meet the Superproject
  - The eternal (well, quite) enemy: change
- Dodging the problem
  - Change management processes
  - Integrated tools
- Facing the problem: Traceability
  - Project meta data
  - Use of meta data
  - Any tool support? Yes, Ophelia!
  - Metadata overuse analysis
- Summary

Presentation outline.
Change management processes

- Paper-driven change management processes
  - Pros:
    - Assigning responsibility to employees
    - Each change is well documented
  - Cons:
    - Information overload (each change is well documented)
    - Amplifies short-sightedness (write make employees).
### Change management processes

- Agile processes (eXtreme Programming and such)
  - Kent Beck’s “Embrace change”
    - Code sharing, rotation of developers, pair programming, self-explanatory code, frequent unit testing and integrations, refactoring
- Pros:
  - People have excellent knowledge about the system (short-sightedness)
  - Code base usually easily modifiable
- Cons:
  - Difficult to apply to large projects, lack of any documentation apart from the code may become an issue.
Integrated tools

- Examples
  - Rational XDE
  - Eclipse (not quite, but maybe)
- Pros:
  - Automatic or semi-automatic support for change management activities
  - Mostly requirements-to-code oriented (vertical)
  - Usually shipped together with well established processes (i.e. RUP)
- Cons
  - Expensive
  - Vertical tracking of change/relationships
Are there any other alternatives?
The roadmap

- Software development problems
  - Meet the Superproject
  - The eternal (well, quite) enemy: change
- Dodging the problem
  - Change management processes
  - Integrated tools
- Facing the problem: Traceability
  - Project meta data
  - Use of meta data
  - Any tool support? Yes, Ophelia!
  - Metadata overuse analysis
- Summary

Presentation outline.
Vertical project tracking
In most cases, tracking relationships is done from requirements to other artifacts…
## Project meta-data

- In any project, there are many more relationships than just those derived from requirements.
- Examples:
  - Bugs to code
  - Documentation to code
  - Code to documentation
  - Module/ component dependencies (design)
  - ...

| Slajd 21 |  |  |  |  |
Traceability is a meta-layer of information, which is present in any project, but not always shown explicitly.
It is much more convenient if artifacts in the project are represented as uniform elements (no matter where they reside, or what tool was used to create them)
Benefits of having meta-data

- Tracing the graph of relationships (get the BIG picture)
- Change impact browsing & estimation
- Complex metrics, involving all software and non-software factors (i.e. is number of words in documentation correlated with the source code? What is the average number of project components «documentation, bug fixes, release notes, etc» affected by a single change in requirements?)
## Benefits of having meta-data

- Change notifications
  - Propagated along relationships in meta-data
  - Ability to build automatic synchronization tools (design : code?)
- Rule-based workflow process definition on top of traceability layer?
  - For instance: a change in a class must result in a new version of its documentation
**Benefits of having meta-data**

- Basic information about any element of the project
- No need to install all software required to open a given artifact
Potential problems with meta-data

- Meta-data must be itself maintained and may become an overhead instead of help
- Be aware of the saturation point
Figure 2 - The value derived from automation versus the granularity of the artefacts in the environment
Wait, if traceability is so great, why is it not present in the development tools?!
• Widespread proprietary tools & formats are hard to integrate
• Limited support from tool vendors
• Emerging integrated environments are closed to extensions, usually sold from one company (Rational), or have not yet reached a decent level of maturity (Eclipse)
• No formal processes of using project metadata exist
So…
Is traceability-enabled environment ever going to be built?
• A uniform interface to all development tools involved in a project development must be available
• OpenSource movement has been very active and the only thing several already available tools need is bringing them together

• Several attempts already available:
  • Eclipse
  • Genesis (EU funded)
  • Ophelia (EU funded)
Introducing Ophelia

Open Platform and Methodologies for Development Tools Integration in a Distributed Environment

http://www.opheliadev.org

http://ophelia.cs.put.poznan.pl
What is Ophelia?

- Ophelia defines a set of standard interfaces for accessing types of tools used in a software project:
  - Bug tracking
  - Project management
  - Documentation management
  - Code repository
  - Etc.

- Ophelia also specifies common, abstract services:
  - Traceability (notifications, relationships)
  - Knowledge (unified access to project artifacts)
What is Orpheus?

- Orpheus is a pilot implementation of Ophelia interfaces
- Supports Open-source and proprietary tools
  - Microsoft Project
  - Bugzilla
  - ArgoUML
  - ...
- Alpha release has initial support for traceability and integrators
Orpheus in action – ArgoUML to Modelling Module
This is an AVI movie showing Orpheus in action.
This is an AVI movie showing Orpheus in action.
**Orpheus in action** – DRE – requirements server
This is an AVI movie showing Orpheus in action.

xDRE is an open-source free requirements management tool, see demo at

http://ophelia.cs.put.poznan.pl/xdre

or write to

Krzysztof.kowalczykiewicz@cs.put.poznan.pl for further info
Orpheus in action – Traceability
This is an AVI movie showing Orpheus in action.
<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Traceability is a project-wide, unified view on all software and non-software elements of the project</td>
</tr>
<tr>
<td>• Traceability allows creation and browsing of relationships among those elements</td>
</tr>
<tr>
<td>• Traceability provides some high level services, not available otherwise, such as change propagation, notifications or metrics</td>
</tr>
<tr>
<td>• There are tools being developed to support traceability entirely, or in parts</td>
</tr>
</tbody>
</table>
Thank you

Questions?

Further questions, please e-mail to:
dawid.weiss@cs.put.poznan.pl