The i-Know Conference
October 11-12, 2017 – Graz, Messe Congress

OpenReq: Recommender Systems in Requirements Engineering

Alexander Felfernig, Martin Stettinger, Andreas Falkner,
Muesluem Atas, Xavier Franch, and Christina Palomares

Presented by: Stefan Reiterer
Institute for Software Technology (IST)
Graz University of Technology, Austria
E-Mail: reiterer@ist.tugraz.at

Outline

- Introduction and challenges
- OpenReq scenarios
- Recommendations for single users
- Recommendations for groups
- Dependency detection and conflict management
- Bid management example
- Conclusions

Horizon 2020 Project OpenReq

- **OpenReq**: Intelligent Recommendation Decision Technologies for Community-Driven Requirements Engineering
- **Duration**: 2017 – 2019
- **9 Partners**:

Hitec Ev.
Germany

Technical University of Graz
Austria

Engineering
Italy

UPC
Spain

H3g
Italy

Vogella GmbH
Germany

Siemens
Austria

University of Helsinki
Finland

The Qt Company
Finland

- **Goal of TU Graz**: supporting RE with single and group recommender technologies as well as conflict detection mechanisms

Introduction

- **Requirements** represent *decision alternatives or* commitments concerning the functionality and quality of the software or service
- Requirements Engineering is a key activity in ICT projects
  - **poor RE is a major risk for project failure**
- High quality requirements are the essential precondition for the success of an ICT project

→ RE receives rarely more than 2-4% of overall project efforts[1]

→ Defect cost fixing: 70$ (RE phase) to 14.000$ (production)[1]

Challenges

1. **Stakeholder and user involvement**: high diversity and number of stakeholders
2. **Software complexity**: Size and complexity of today's software continuously increases
3. **Stakeholder heterogeneity and distribution**
   different backgrounds of stakeholders: management, technical, legal etc.
4. **Evolution and change frequency**
   shorter time to markets, demanding knowledgeable customers, shorter release cycles

OpenReq Scenarios: „bid management“

Scenario: Request For Proposals (RFPs) management for railway safety systems
→ several hundred pages of natural language requirements

Goals

1) identify and extract requirements
2) Reuse technical decisions made in previous projects
3) Assigning requirements to stakeholders
4) Support group decision making

OpenReq Scenarios: „cross platform open source“

- Scenario: community of individuals and professionals contributing to projects

Goals

1) Advanced user engagement (identifying potential contributors)
2) Internal release planning (detecting urgent requirements)
3) Management of requirements (e.g. detecting requirements dependencies)
4) Detection of new requests (e.g. based of community discussions)

OpenReq Scenarios: „telecommunication scenario“

Scenario: improving the RE process as well as reacting to opinions of customers

Goals

1) Identify and extract requirements from user requests
2) Monitor the communities to identify issues
3) Propose prioritization indicators for requirements e.g. from user discussions
4) Support stakeholders in the preparation for a group decision (e.g. by highlighting relevant topics)

Single user recommendations

- **Collaborative filtering**
  →Recommending requirements to review based on requirements that have been reviewed by Stakeholders with *similar interests*
- **Content-based filtering**
  → *Recommending similar requirements* from previous projects that are relevant for a new project
  → *Recommending similar stakeholders* from previous projects that are relevant for a new project

Single user recommendations

- **Knowledge-based recommendation**
  → Release planning: Requirements should be implemented in certain releases with regard restrictions and dependencies

- **Goals regarding single user recommendations**
  → In depth integration of these technologies

Recommendations for groups

- **Release planning**
  → Consensus in the group among the planned releases
  → Inconsistencies can occur e.g. when assigning requirements to releases

- **Requirements evaluation**
  → Effort in MMs, risk level, potential turnover, and importance of implementation

Recommendations for groups: Conflict management

- **Conflicting preferences**
  - e.g. regarding the meta-properties (effort, risk etc.) of requirements
  - e.g. regarding the assignment of requirements to releases
- **Solutions**
  - Identify the minimal changes that are necessary
  - Applying social choice functions (aggregation functions) → e.g. least misery: recommendations that don‘t ignore negative recommendations

Recommendations for groups: Goals

- **Goals regarding group recommendations**
  - Gain in depth insights into group processes in RE and improve them
  - Identifying occurring biases and counteracting if necessary (e.g. GroupThink)
  - Achieve consensus among stakeholders

Example: Bid Management

- *Use case:* Large-scale industrial systems related to the Siemens Mobility division
- *Request For Proposal* (RFPs) e.g. for railway safety systems are issued by national railway providers
- *Sales departments* deliver proposals in *natural language* (several 100 pages MS Word documents)

Example: Bid Management

- *Various kind of requirements* like domain specific, physical, non-functional, references to standards and regulations, etc.
- *Several subsystems*: signaling hardware, track indication, interlocking software etc.
- *Different stakeholders*: project manager, a requirements administrator, a system architect, technical experts etc.

→*Goal of the RE process:* Technical compliance has to be ensured

Example: Bid Management

- Natural language text is imported into Polarion ALM
- Recommendation approaches can support in the following:
  - Separating real requirements from explanatory text
    → with classification based on domain knowledge and experience from past bid projects
  - Assigning requirements stakeholders for evaluation
    → Recommendation can suggest the corresponding stakeholder roles
  - Evaluating the requirements for technical compliance
    → Recommendation is based on similar requirements.

Conclusion

- Support and enhance existing tools with new technologies
- Build a basis for new RE tools to
  - improve processes
  - simplify maintenance
  - increase quality
  - advance group decision support
- Provide a *showcase* for demonstrating the capabilities of all the OpenReq components

THANK YOU!

References

[1] “Hype Cycle for Application Development, 2015.” [Online]. Available: https://www.gartner.com/doc/3101023/hype-cycle-application-development-. [Accessed: 01-Apr-2016].