Modeling requirements artifacts in an IDE

Master thesis, final presentation, FS2020

Author: Robert Niemiec

Supervisors: Nitish Patkar, Nataliia Stulova, Andrei Chiș
Requirements

Establish a common vision of the product
Requirements

Establish a common vision of the product

Act as a contract
Requirements

- Establish a common vision of the product
- Act as a contract
- Various artifacts and formats used
Requirements

Documents

Physical

Software
Requirements

Documents
Physical
Software

Fragmented knowledge
Requirements

Documents

Physical

Software

Fragmented knowledge
Dynamic management
Centralized Requirements Management

- Integrate project artifacts into a single platform
- Model a selection of artifacts
- Workflows:
  - Creation, updating, removing artifacts
  - Navigation
  - Visualization
What are the most commonly used requirements formats and related artifacts within the project development process?
RQ1 – Requirements Artifacts

- Analysis of RE literature
- Extraction of mentioned artifacts
- Compilation of a list of artifacts
RQ1 – Requirements Artifacts

- Analysis of RE literature
- Extraction of mentioned artifacts
- Compilation of a list of artifacts
RQ1 – Requirements Artifacts

- Analysis of RE literature
- Extraction of mentioned artifacts
- Compilation of a list of artifacts
RQ1 – Requirements Artifacts

- Analysis of RE literature
- Extraction of mentioned artifacts
- Compilation of a list of artifacts

62 artifacts obtained
Research Questions

RQ1 What are the most commonly used requirements formats and related artifacts within the project development process?

RQ2 What are the main characteristics of these artifacts?
RQ2 – Artifact Properties

Classification of collected artifacts

Analysis of results

Extraction of findings
RQ2 – Classification Dimensions

1. **Format**
2. **Nature**
3. **Contains**
4. **Helps Create**
5. **SDLC Phase of Origin**
6. **SDLC Phase of Use**

**Format**
Textual/Graphical/Mixed

As a `<user>`, I want to `<perform action>`, so that `<goal>`

User Story – Textual
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Format

Textual/Graphical/Mixed

Sketch - Graphical
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Format

Textual/Graphical/Mixed

UML Diagram - Mixed
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Nature
Digital/Physical

User Story
As a potential customer
I want to read book reviews
So that I can decide which one to buy

Story Card - Physical
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Nature
Digital/Physical

UML Diagram - Digital
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Contains

Other artifacts

User Story Map – Example

Story Map – contains user stories
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Helps create
Other artifacts

Wireframes → Mockups → Prototypes
RQ2 – Software Development Life Cycle (SDLC)

Requirements

- Requirements are elicited, collected and specified
The requirements are reasoned about; a solution is designed

- Development process is structured
RQ2 – Software Development Life Cycle (SDLC)

Development and Testing
- The solution is developed
- Source code is created and tested
RQ2 – Software Development Life Cycle (SDLC)

Deployment and Maintenance

- The solution is deployed
- Maintenance activities (e.g. bug fixing, usage reports)
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

Mind Map – Requirements phase
RQ2 – Classification Dimensions

1. Format
2. Nature
3. Contains
4. Helps Create
5. SDLC Phase of Origin
6. SDLC Phase of Use

SDLC Phase of Use

Release Plan – Development and Testing phase
RQ2 – Artifact Properties

Classification of collected artifacts

Analysis of results

Extraction of findings
RQ2 – Artifact Properties

Classification of collected artifacts

Analysis of results

Extraction of findings
RQ2 – Observations and Findings

Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.
RQ2 – Observations and Findings

Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.

62 artifacts extracted
RQ2 – Observations and Findings

Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.

62 artifacts extracted

Stakeholder variety
RQ2 – Observations and Findings

Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.

62 artifacts extracted
RQ2 – Observations and Findings

Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.

62 artifacts extracted
Finding 1: A plethora of artifacts, with varying characteristics, are available to practitioners; these artifacts need to be easily accessible to stakeholders.

62 artifacts extracted

Stakeholder variety  Supported activities  Tools and media

Harder to maintain the „big picture“ of a project
Finding 2: If artifacts are meant to be used by developers and designers, then they should be clear for these groups.
RQ2 – Observations and Findings
RQ2 – Observations and Findings
RQ2 – Observations and Findings

Finding 2: If artifacts are meant to be used by developers and designers, then they should be clear for these groups.
Finding 2: If artifacts are meant to be used by developers and designers, then they should be clear for these groups.
Finding 2: If artifacts are meant to be used by developers and designers, then they should be clear for these groups.

Vague and unstructured requirements are problematic for developers

Structure needs to be applied to requirements

Singular requirements management platform
Finding 3: Reflecting requirements changes across artifacts can be challenging.
RQ2 – Observations and Findings

Finding 3: Reflecting requirements changes across artifacts can be challenging.

Requirements are subject to change  Changes have to be reflected across all artifacts
Finding 3: Reflecting requirements changes across artifacts can be challenging.
Finding 3: Reflecting requirements changes across artifacts can be challenging.

- Requirements are subject to change
- Changes have to be reflected across all artifacts
- Inconsistencies likely to happen in distributed settings
- Singular requirements management platform
RQ2 – Observations and Findings

Finding 4: Artifacts have relations and dependencies; gaining an overview of the structure is important for understanding the complete requirements picture.
RQ2 – Observations and Findings

Finding 4: Artifacts have relations and dependencies; gaining an overview of the structure is important for understanding the complete requirements picture.

Artifacts often exist in hierarchies.
RQ2 – Observations and Findings

Finding 4: Artifacts have relations and dependencies; gaining an overview of the structure is important for understanding the complete requirements picture.

Artifacts often exist in hierarchies

Linking of artifacts is cumbersome; lack of clear guidelines
RQ2 – Observations and Findings

Finding 4: Artifacts have relations and dependencies; gaining an overview of the structure is important for understanding the complete requirements picture.

Artifacts often exist in hierarchies

Linking of artifacts is cumbersome; lack of clear guidelines

50% of the artifacts can help create others (e.g. Epics to use cases)
RQ2 – Observations and Findings

Finding 4: Artifacts have relations and dependencies; gaining an overview of the structure is important for understanding the complete requirements picture.

Artifacts often exist in hierarchies

Linking of artifacts is cumbersome; lack of clear guidelines

Clear linking functions are needed
Research Questions

**RQ1** What are the most commonly used requirements formats and related artifacts within the project development process?

**RQ2** What are the main characteristics of these artifacts?

**RQ3** What advantages do we gain if we specify and manage artifacts within a single platform?
Centralized Requirements Management

- Integrate project artifacts into a single platform (IDE)
- Model a selection of artifacts
- Workflows:
  - Creation, updating, removing artifacts
  - Navigation
  - Visualization
Moldable Requirements Manager (MReM)

- Tool for modeling and managing requirements
  - Based on Pharo, GToolkit
- Visual overview of the artifact structure
- Custom views, workflows for artifacts
- Linking requirements and source code
Live Demo
Future Work

Modeling further artifacts
Future Work

- Modeling further artifacts
- Different visualization schemes
Future Work

- Modeling further artifacts
- Different visualization schemes
- Support for data formats (e.g. ReqIF)
Questions?