VISDOM

Visual software diagnostics

D1.2.1 Detailed definition of visualization use cases

Programme	ITEA3		
Challenge	Smart Engineering		
Project number	17038		
Project name	Visual diagnosis for DevOps software development		
Project duration	1st October 2018 – 30st June 2022		
Project website			
Project WP	WP*1* - Pre-studies and requirements		
Project Task	Task *1.2* – Specification of visualization requirements		
Deliverable type	X	Doc	Textual deliverable
		SW	Software deliverable
Version	V06		
Delivered	14/01/2020		
Access		Public	
	x	Abstracts are public	
		Confidential	

Document Contributors		
Company	**Author**	**Role**
EXPERIS	Ester Sancho	writer/editor
EXPERIS	Gema Maestro	writer
GRO	Paris Avgeriou	writer
OCE	Lou Somers	writer
OULU	Markus Kelanti	writer
OULU	Henri Bomström	writer
TAU	Kari Systä	writer
TAU	Outi Sievi-Korte	writer
UPC	Lidia López	writer
UPC	Xavier Franch	writer
VINCIT	Veli-Pekka Eloranta	writer

Document History			
Date	**Version**	**Editors**	**Status**
23/09/2019	ToC	EXPERIS	Table of Content
06/10/2019	V01	EXPERIS	Draft
22/10/2019	V02	TAU	Draft
05/11/2019	V03	EXPERIS/TAU	Draft
26/11/2019	V04	GRO/OCE	Draft
09/01/2020	V05	VINCINT	Draft
13/01/2020	V06	EXPERIS/UPC	Final Version
14/01/2020	07	EXPERIS	Submitted Version

Table of Contents

Table of Figures

1. EXECUTE SUMMARY

VISDOM project aims at developing new types of visualizations that utilise and merge data from several data sources in modern DevOps development. The final goal is to provide simple "health check" visualizations on the status of the development process, software and usage.

For this purpose, WP1 aims at gathering requirements for visualizations in software projects based from multiple sources at different software lifecycle phases. An important driver for establishing the requirements is to identify and define the use cases that VISDOM will use as demonstrators.

This deliverable presents the analysis that has been carried out within T1.2 in order to establish and set all the different use cases that will be contemplated within VISDOM lifetime.

There 3 main use cases based including quality aspects, SaaS-business and visualization in teaching. Anyhow due to the different perspectives and companies working on the quality aspect use case, there have been described additional sub use cases.

The methodology used for the analysis and definition of these use cases is quite similar and described in Section 3. Secondly, the problems aimed to be solved were identified and can be consulted in Section 4. Finally, each use case is presented, describing its actors and stakeholders involved, specific business needs and user's perspective, together with the high level requirements that came out after this analysis.

2. INTRODUCTION AND PROJECT BACKGROUND

The main objective of WP1 is to gather requirements of real-time in software projects. The visualizations are based on data aggregated from software product, tools, infrastructure and repositories at different phases of software lifecycle. Visualization requirements are gathered from the start of the project and continued for at least eight months through state-of-the-art and -of-practice analysis.

Developing visualizations of software project in DevOps context requires not only the specification of new features but also determination of appropriate data sources, integrators, processing algorithms and graphical representations from existing ones or alternatively implement their extensions. For this purpose, state-of-the-art analysis is mainly conducted in the areas of visualisation, data mining and processing in software projects considering that analysis of DevOps has previously been investigated by VISDOM partners. A new contribution shall be in determining appropriate techniques and methods to implement visualization in the context of DevOps where cross-discipline collaboration and rapid development in software projects are emphasised.

Deliverable D1.1.1, conducts the state-of-the-art analysis, in parallel with state-of-practice investigations starting with all use cases described in this deliverable (visualisations contexts) where VISDOM partners have identified the needs for real-time visualisation based on aggregated data from different sources. Given that VISDOM has experts in software visualisation, both in universities and companies, conducting state-of-practice investigations will help in the selection and determination of extensions to be implemented in the visualizations during their development in WP2-WP3.

T1.2 focuses on defining requirements of the visualization's designs. This involves concrete definition of the initial use cases of VISDOM demonstrators. The use case definitions will prioritize the technical work on proof-of-concept and exploitation - dissemination.

3. USE CASE DEFINITION METHODOLOGY

For the definition of the diverse use cases covered by VISDOM, a previous and exhaustive research was required. This research was led by the universities involved in the project and it was performed at national level. Therefore, OULU University together with TAU have developed the use cases for Finnish companies, GRO for the Netherlands company OCE and UPC has collaborated with Experis in the definition of the Spanish use case.

As similar methodology was used for defining all the VISDOM use cases, regardless the country. All methodologies used are based on the single-case study [Runeson et al. 2012].

Below is an explanation on how each country has carried out this study.

Use Case Definition in The Netherlands

The research method used to derive the Quality use case is the Case Study. Specifically, the case study design reflects an embedded, single-case study [Runeson et al. 2012], i.e. we study multiple units of analysis contained within the same case. Our case is Océ: a large, high-tech company which is specialised in the production of printers. Focusing our study on a single organisation allowed us to answer our research questions in a high level of detail and understanding. The units of analysis are the practitioners that have a long, hands-on experience with dealing with Technical Debt and Quality issues.

The development of embedded systems in the case company is interdisciplinary, i.e. software development involves dealing with other engineering disciplines. Software development in this company also follows agile practices (e.g. a backlog is maintained for new requirements). Moreover, they have systematic methods to apply architectural rules among software engineers in the whole organisation. The architectural rules are documented in a reference architecture. Software team members within the organisation have specific roles: technical leader, software architect, and software engineer. Technical leaders and software architects are the main design decision makers within the software team.

Figure 1 shows an overview of data collection and data analysis. Our data collection process followed three main steps (shown in Figure 1):

1) Background knowledge: In this step, we asked practitioners to answer questions about their experience, and especially architectural experience.
2) Individual interviews to gather architectural design scenarios: In this step, we interviewed practitioners individually and asked them about design scenarios that involved technical debt and quality issues. Interviews provide the best means to determine experience of software engineers and the insights they obtain from these experiences.
3) Focus group to reflect on meaning: After interviewing each practitioner individually, we met with the five practitioners in a plenary session for a focus group. Our main goal from the focus group was to use group dynamics to support exchanging direct opinions between the practitioners.

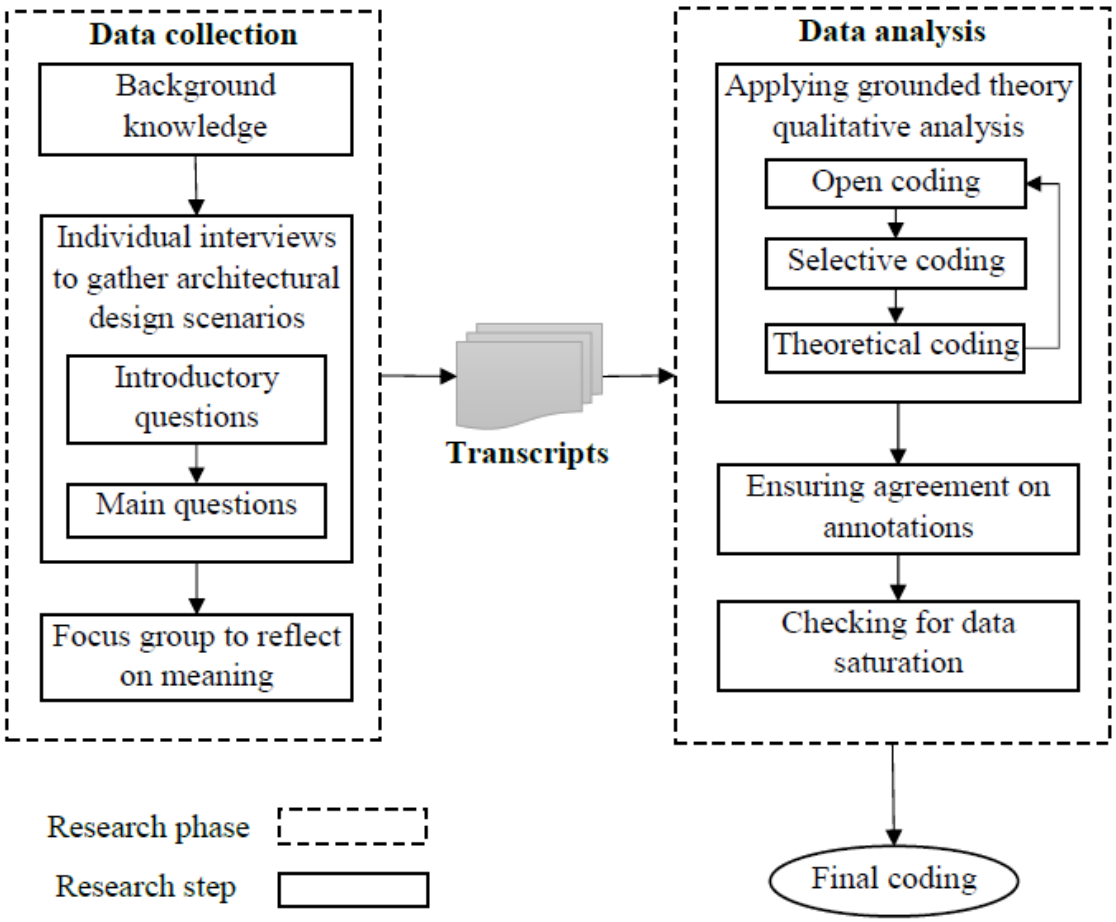

Figure 1: Data Collection & Data Analysis

The data analysis process consisted of three main phases (as shown in Figure 1):

1) Applying grounded theory qualitative analysis: To analyse the transcripts of the interviews and focus group, we followed an iterative qualitative data analysis process based on ”Grounded Theory”.
2) Ensuring agreement on annotations: In qualitative content analysis research methods, reliability of the analysis is critical. To strengthen reliability, two researchers conducted two iterations of reliability tests: The first researcher independently annotated selected sentences (that were also coded by the second researchers) and compared it to the original annotations of the second researcher; disagreement was discussed and the groups of annotations were modified until consensus was reached.
3) Checking for data saturation: in the first phase of the analysis, we analysed the some of the interviews and the focus group to reach a mature set of concepts. In this phase, we deductively annotated each sentence of the remaining interviews using our final codes. Our main goal was to check if we are able to find new concepts or we can only find the same concepts as the previously annotated transcripts. Our results show that we could not find any additional concepts in the fifth interview. In other words, we reached data saturation.

Use Case Definition in Finland

Data collection and analysis for Finnish company use cases were designed as case studies based on Runeson et al. [2012]. Based on the VISDOM objectives, the need to identify the

requirements for the visualisation and identify the first visualisation prototypes, the process started with workshops. The goal of the workshops was to clarify the company use cases with the company representatives and identify initial visualization needs and concrete visualisations to research, design and implement in the project.

After the workshop, interviews were arranged to support the requirements anlaysis for the company use case and selected concrete visualisations. 4-5 interviewees were selected from each case to give an initial representation of stakeholders for the selected visualisation.

A semi-structured interview template was designed based on the workshop results and project goals to assist the interview. This template was peer-reviewed by researchers and also by company representatives to check if the terminology used in the template was clear and understandable for the company stakeholders.

Each interview is scheduled to last 1 hour and the interview session is attended by at least 2 researchers interviewing the company stakeholder participating the session. The session is recorded and later transcribed by a third party. A short summary is also written from the session and given to the interviewee in case there are mistakes that the interviewee can correct or the interviewee wants to add additional information to supplement the summary. If no response is received after 2 weeks, the summary is assumed to be accurate.

Both summary and transcription will be analysed thematically, based on the interview themes and any other theme that naturally surfaces from the material. The result of the analysis are used to formulate the initial visualisation requirements for the selected concrete visualisation for each use case and overall requirements for the Visdom dashboard and visualisations.

Use Case Definition in Spain

In Spain, the design of the use case was also based on Runeson et al. [2012]. The UPC, subcontracted for the development of this project, made available an innovative tool developed in the framework of the Q-Rapids project, a European Union funded project which provides a data-based approach to monitoring and analysing quality in the context of rapid software development.

Firstly, the suitability of using this tool for the project was analysed, obtaining very satisfactory results. Next, it was considered whether this tool could be applied to internal projects developed by Experis, to produce an advanced version of it based on the general objectives of VISDOM and on the specific needs of these projects.

For this purpose, a workshop was conducted to present the project and the tool to different project managers working within Experis. After the presentation, individual interviews were conducted following an interview template designed hand in hand with UPC and trying to obtain relevant information from the stakeholders in order to identify different needs that could be covered by VISDOM solution and transform them into requirements that should guide the development during this project.

The workshop as well as the interview were conducted internally by the innovation department of Experis to the corresponding stakeholders involved. Afterwards UPC transcribed the all the feedback and interview results for its subsequent analysis. As a result we obtained the main requirements that will guide VISDOM project development for this specific use case (see Section 5.2).

4. THE PROBLEM STATEMENT

Problem 1	
The problem of	Having to take into account large amounts of data / variables at different levels to monitor the quality of SW development processes or other projects
Affects	Project managers and product developers, as well as the quality of the SW developed or efficiency of the process.
The impact of which is	Lack of knowledge about the origin of problems/errors in the development.
A successful solution would	Identify and visualize the key quality indicators for a specific project/development, using heterogeneous data coming from various available sources, as well as aggregated by organizational level hierarchies.
Problem 2	
The problem of	Accumulating Technical Debt
Affects	Quality and maintainability of embedded software
The impact of which is	That users may be confronted with unexpected errors or behaviour and that software developers have to spent increasing amounts of time in repairing this.
A successful solution would	Identify and visualize the existing technical depth using data from various data sources.
Problem 3	
The problem of	Providing teachers actual information about students' progress in an efficient way
Affects	Teachers and students
The impact of which is	Teachers are not aware of the real situation and thus spend time and effort on wrong things. Students do not get help when they need it.
A successful solution would	Improve teaching results, give personal support, and optimize resource usage - especially in case of on-line teaching.

5. USE CASE 1: VISUALIZATION OF QUALITY ASPECTS.

5.1. Introduction

The quality aspect in the context of DevOps has two facets. First as the internal quality of a software product and its associated process. The second facet of quality is the user perceived quality of the product based on user interactions. Visualizations required for the two facets are complementary in their goals, sources of information and are valuable for different stakeholders. The quality of the software and the associated process is primarily relevant for technical management and is based on artefacts that are linked to source code but also the running system: issue tracking, log files, project management data, performance traces, ecosystem dependencies, etc. In this context, the goal is to provide high-level and focused visualizations that would allow the development teams to improve their development process and the quality of their code (e.g. technical debt, opportunities for refactoring). An added value of these visualizations is the linking between design-time entities (e.g. quality issues at the source code level) with run-time entities (e.g. operational bottlenecks).

Several partners will be contributing to this use case. Qentinel is developing an analytics solution to visualize the whole DevOps cycle, linking the internal quality and external quality in to the same metrics and providing a way to forecast these metrics per release candidate. The latter will enable identification and prediction of potential failures and errors. The quality of the product as perceived by the users is primarily relevant for business management and can be inferred in a data driven way based on telemetry information. In this context, the goal is to understand the various patterns of usage of the deployed software in order to help steer the future development directions. OCE has a particularly valuable case study with large amounts of telemetry information from deployed cyber-physical systems. One special use case that will be developed will be addressing the problem of visualization not only single project information but also an overview of all the projects in the company-wide ecosystem. Experis in collaboration with UPC will also participate, by implementing, adopting and improving a new tool recently developed for providing SW development analysis, as well as developing specific metrics or indicators relevant for different stakeholders.

5.2. UC1.1 Sub Use Case by EXPERIS

Experis IT is a leading company in providing IT solutions to meet any need in the field of Information Technology and Communication: consulting and compliance, headhunting technical assistance, software development, project management, quality assurance services and outsourcing of services both nationally and internationally.

As already mentioned, we plan to implement Q-Rapids into particular projects in order to develop an advanced version meeting the daily problems our stakeholders are facing, as well as the VISDOM objectives which are mostly shared.

5.2.1. Actors & Stakeholders

The main actors and stakeholders of this particular Use Case are principally Product Managers and Product Owners, as well as developers and testers. All of them are involved in the development of different processes of faces of a product or service delivery and will be

interested in different type of information or granularity of information regarding the whole project's life cycle.

- Product Manager: responsible for the strategy, planning, execution and launch of a Product. His mission is to coordinate all the teams involved, as well as to monitor the management of incidents, complaints or improvements that need to be made to it.
- The role of Product Manager (PM) includes coordinating different departments, such as UX + Design + Programming, requiring updated and preferably easy-to-understand information/indicators on each department.
- Product Owner: The Product Owner is responsible for ensuring the team brings value to the business. It represents the internal and external stakeholders, so it must understand and endorse the needs of all users in the business, but also the needs and performance of his team.
- Developers & Testers: The aim of testers and developers is the same i.e. to provide a quality product to the stakeholders.

5.2.2. Business Needs

The technology transformation is affecting all different kind of economy sectors little by little. Some of our clients come to us in order to accompany them on this path. Within this technical transformation software becomes of highly importance and has a great impact on the business itself. Many of our clients will be benefited by a system that enables to collect data from almost any source available and aggregate them according to some business requirements and on complexity level.

5.2.3. Users' Perspective

The main users in this use case are product managers and product owners. These users showed a particular interest in the following functionalities:

- Facilitate the transfer of information between departments.
- Provide relevant information visualised in an easy-to-understand manner.
- Combine data from quality process and quality product.
- Provide a dashboard that facilitates decision making at a glance. In addition, it allows a deeper analysis of the problems identified.
- This panel should also show predictions on key decision areas and "what if" simulations that are considered very useful for future decision making.

5.2.4. Demonstrator Specifications & Requirements

Demonstrator Title	Quality Assurance
Author/company	Ester Sancho, Gema Maestro / EXPERIS IT Lidia Lopez, Xavier Franch / UPC
Partners involved	EXPERIS, UPC,

Primary Actors	Product Managers, Product Owner, Developers, Testers, QA Consultants

Requirement	Description	Priority (high/medium/low)
R-UC1.1-01	Enable the configuration of what can be measured	High
R-UC1.1-02	Possibility to personalize the indicators according to the needs of the user	Medium
R-UC1.1-03	Possibility to define indicators that combine quality product and quality process aspects.	High
R-UC1.1-04	Provide help for decision making for Product Owners/ Product Managers for managing resources	High
R-UC1.1-05	Provide easy-to-understand visualization of indicators, transparent for the user type.	Medium
R-UC1.1-06	Provide near-real-time updated indicators	High
R-UC1.1-07	Simulations of “what if” scenarios	Medium
R-UC1.1-08	Highly compatible with all types of data sources	High

5.3. UC1.2 Sub Use Case by QENTINEL

Qentinel is a SME company that focuses on quality intelligence and robotic testing. The company’s long-term goal is to expand their international business by leveraging research innovations into cutting edge automated testing and quality assurance products. Their products offer DevOps teams a clear picture into their processes and provides insight on possible bottlenecks. Qentinel’s aims to reach this goal by validating and expanding their DevOps metrics portfolio. Furthermore, Qentinel aims for specific metrics, such as code quality and architectural complexity, and other new metrics that further enhance their products. Additionally, Qentinel wishes to improve its forecasting abilities by adopting machine learning features.

5.3.1. Actors & Stakeholders

The primary actors are developers, users, managers and customers. Developers can specialize as testing engineers and architects. Users represent the persons that use the company’s products, both internally and within customer companies. Managers can be divided into sales and development related roles, where development issues are managed by project and product owners. Finally, there are the customer companies that use Qentinel’s products.

Qentinel’s use case focuses mostly in supporting managerial decisions based on insights from progress and improvement. From developer viewpoint, the most important information relates to release quality. Qentinel development pipeline is highly automated and can be deployed

without managerial intervention. However, managers do make production related decisions on larger issues. Additionally, managers – especially sales related roles – communicate with customers and map their issues and needs. Customers can produce feedback, feature requests and bug reports for developers. Developers process these tickets in sprints.

5.3.2. Business Needs

The benefits of creating a dashboard begin with supporting the decision-making process on different levels of development. Role-specific issues and problems can be visualized, and input provided for resolving them, allowing people to focus on the things that truly matter. However, mere identification of problems is not enough as problems should be caught before they manifest. Potentially problematic situations can be identified, and possible evasive measures executed based on managerial judgement. Furthermore, the dashboard can act as an extension for reducing the need for communication between different stakeholders.

From a more sales point of view, Qentinel's customer projects are somewhat similar. Analytics on project lead times from start to production would allow predictions on budget, the development effort required, and possible problem situations that might arise during development. Data could be collected from the discussions between sales and customers and might result in quicker revenue cycles and faster delivery to clients. Furthermore, sales progression visualizations could allow the company to trigger preparations for planning activities beforehand.

From a customer communication perspective, Qentinel could profit from visualizing cumulative value and the AI assisted release prediction for clients. Currently, the best way for acquiring direct feedback is through silent knowledge: by talking to people and forming ad hoc evaluations of the situation. Furthermore, a customer journey could be helpful to understand clients: are the customers leaving? If then why? Are they happy? How could they be happier and stay longer?

5.3.3. Users' Perspective

The main users in Qentinel's case are managerial level employees. These users wish to have a dashboard that would support their decision-making process by displaying the current situation, predictions over key decision points and an explorative view for further analysis. The aim is to build a dashboard that results into action from one glance. In order to debug production related issues, the users should be able to set their own thresholds for problem cases. However, the display should not contain too much information to reduce analysis paralysis.

Creating a dashboard allows the company to see the big picture with the possibility of zooming in to single problem cases. The dashboard should include visualizations that point to each role's current issues and include information on how to resolve them. These views should include metrics from external tools to pinpoint problems and allow transitions to these tools. Currently, problems can be seen when they have already manifested. A better way would be to try and identify situations that could lead into problems, such as anomalies in production or changes in trend. Trends can be further used to represent how goals are reached. Identifying the top leading causes for trend changes, in good or bad, would allow the users to better understand cause and effect in production. Furthermore, a goal reaching forecast, or a sprint prediction would allow managers to react fast and steer development before problems occur.

Qentinel's development efforts are divided into sprints. One of the most important decisions managers make is deciding what goes into the backlog and into a release. Furthermore, the assessment and prediction of workload is an important factor in delivering software as releases. Even inaccurate predictions can convey information that either the task was too difficult or the ability to predict was subpar.

From a testing perspective, the main concern is that the bug reports lack information. The testers need enough information to get the job done: what are the steps to reproduce the bug and what is the goal of the ticket. Overall, testers would want to see their task backlog, a list of bugs and their severity, and feature requests. If possible, they would benefit from seeing how users interact with the user interface.

5.3.4. Demonstrator Specifications & Requirements

Demonstrator Title	Dashboard for supporting quality intelligence and value creation.
Author/company	Henri Bomström, University of Oulu
Partners involved	Qentinel, University of Oulu
Primary Actors	The primary actors in this case are the process and sales managers.

Requirement	Description	Priority (high/medium/low)
R-UC1.2-01 Fault indication	The dashboard indicates (potential) problems based on the user's role or configuration. The system provides a suggestion on resolving the issue.	High
R-UC1.2-02 Deployment readiness	The system indicates readiness for deploying changes to production. This decision can be made by a developer on a "good enough" level.	High
R-UC1.2-03 Recommendations	The system provides recommendations for future actions.	Medium
R-UC1.2-04 Change deltas	Changes in key metrics should be visualized by utilizing delta values. Example includes commits and bugs.	Medium

5.4. UC1.3 Sub Use Case by OCE

Océ develops high-end professional digital printers in three categories:

1) Continuous-feed printers for massive print volumes and fast (1100 A4's per minute), high-quality results in full color or black & white.

2) Highly efficient, high-volume (8500 double-sided A4's per hour) printers for in-house printing or publishing.
3) Large format printers for display graphics (160m2 of glossy banners per hour) and high-quality CAD/GIS applications.

Professional digital printers are expected to be highly productive and customers ideally expect them to work 24x7.

The development of a new printer platform normally takes a number of years, where developers from different disciplines (chemistry, physics, mechanics, electronics and software) are working together. The embedded software which controls the behavior of the physical machine is an essential part of the printer and it is important to ensure its quality to avoid issues when the printers are deployed.

This sub use case is about the monitoring and tracking of the quality of the printer software in the field, when the printers are actually being used by customers.

5.4.1. Actors & Stakeholders

The primary actors in this use case are developers and their managers.

Some examples are R&D project leads (e.g. interested in mean prints between error), domain architects (relations with functionality used by customers), multidisciplinary developers (which errors do occur, what are their root causes), software architects (monitor reference architecture quality), software testers (regression tests overview), service (which machines do need more attention than others), and of course software engineers (how well is my code performing).

5.4.2. Business Needs

The development process of the embedded software at Océ is reasonably state of the art with scrum based sprints, model driven development, version management, automatic builds and regression testing with software-in-the-loop simulation of the printer hardware and mechanics.

A target for the near future is to have faster deployment cycles of new versions in the field, which increases the need for good overviews of the quality of the deployed software.

The printers in the field log a big amount of structured data, for example originating from the sensors in the printer, but also about the status of the software and the usage of the printer. These data can be used to construct overviews of the quality and trigger actions if necessary.

Understanding the various patterns of usage of the deployed software also helps to steer future development directions. Thus, it is not just single project information visualizing but also an overview of the projects in the ecosystem.

5.4.3. Users' Perspective

The main overviews users are interested in are quality indicators of the deployed software, and their relations with software and machine versions and printer usage.

So, the focus is on visualizations that utilize and merge data from several data sources. The aim is to provide a number of "health check" visualizations about the state of the development process, software and use. The visualizations provide early indicators of issues, and in case of suspected problems, the developed techniques enable quick investigations (zooming-in) to possible problems and their root causes.

5.4.4. Demonstrator Specifications & Requirements

Demonstrator Title	Visualizing Technical Debt
Author/company	Lou Somers (Océ) and Paris Avgeriou (RUG)
Partners involved	Océ, RUG
Primary Actors	Technical leader, software architect, software engineer

Requirement	Description	Priority (high/medium/low)
R-UC1.3-01 Data aggregation	Different data sources will be combined such as issue tracking, commits, run-time logs and other tools from the DevOps toolchain.	High
R-UC1.3-02 Data analysis	Data will be analysed using AI techniques (e.g. Machine Learning or Natural Language Processing) to automate the identification and prioritization of Technical Debt items.	Medium
R-UC1.3-03 Technical Debt identification	Items of Technical Debt will be identified and measured using both established and novel metrics. These items will span different levels, from the source code to the architecture.	High
R-UC1.3-04 Technical Debt visualization	The visualization of Technical Debt will include different perspectives: visualization for individual items, for system versions, across evolution.	High

5.5. UC1.4 Sub Use Case by INVENCO

Invenco is a Finnish company focused in Enterprise Performance Management consulting and on its own software products. The company's long-term goal is to further expand their development process to support DataOps types of operations. This goal can be reached by utilizing data-analysis and visualization techniques to support their existing development process. To tackle this task, the goal of Invenco's use case is to create a process tracking tool that was described as "Trello on steroids". Later, Invenco aims to include this product into their portfolio in order to provide monitoring and visualization services for their customers.

Invenco's development process revolves heavily around Trello. Multiple different boards are used to drive the development process where features are added to the production backlog as tickets based on their necessity and urgency. In an optimal case, the tickets would be processed in priority order into releases. However, sometimes unforeseen circumstances, problems in

development, or changes in company strategy can result in new tickets that precede priority over the existing ones. Development issues manifest when tickets remain stationary for long periods of time, their state changes back and forth, or the planned priority order changes noticeably.

The main challenge in Invenco's use case is to detect when their development process needs to be re-evaluated. The proposed solution allows Invenco to identify problems in their development process by offering insights on backlog changes. By examining and re-evaluating problematic tickets, development related issues can be addressed pre-emptively instead of reactively. Additionally, identifying production issues allows Invenco to steer their development process and gain better understanding on which features can be completed for each release, leading to better coordination and improved external and internal communication. Furthermore, the accuracy of workload and monetary estimates can be improved by identifying the effort distinct for different types of tickets.

5.5.1. Actors & Stakeholders

The primary actors in this use case are the developers, managers, and customers. Developers can take on the roles of regular developers, testers, user interface (UI) designers, consultants, and product owners. Additionally, the managerial actors can take on the sales, customer and product representative roles. Finally, there is the customer actor.

Invenco's Oulu branch is developing their part of the InControl platform and other products in teams of two or three. Developers are doing both the "dev" and "ops" at the same time. Currently, the product owner has the final say in daily development decisions. In case the team has to make decisions that affect larger parts of the product, the product development representative is involved in the decision-making process. Product development tasks are created through a formal process as Trello tickets by the product owner or a manager. Developers then process the tickets in priority order. Lastly, customers can relay their needs to the development team through the sales or customer representatives. Alternatively, change requests can also be sent through the product owner.

5.5.2. Business Needs

Trello is the heart of Invenco's development process. Each development task is transformed into a Trello ticket through a formal process and placed on a specific board. Visualizing the Trello-based development process has multiple advantages as it allows product owners and managers to monitor and spot flaws in the development process, estimate release cycles more carefully, and to benchmark development and sales estimates against real development data.

Monitoring the development process and detecting flaws within it allows the company to know if the development process is continuing as planned. The Trello backlog includes only items that have been accepted for development based on necessity and importance. Thus, the priority of tickets is vital for the overall development process. By identifying the problematic tickets, product owners and managers can re-evaluate the development process' direction. Additionally, identifying problems in the development process allows product owners to estimate which features can be completed for each release. In the case of delays, production issues can be communicated to internal and external stakeholders. Furthermore, by examining the throughput of tickets on certain boards the internal stakeholders can deliver better estimations for feature workloads and costs.

5.5.3. Users' Perspective

The main user in Invenco's case is the product owner. From this viewpoint, the user should be able to determine if the development process' task priority has to be re-evaluated. Problems in the process can manifest for various reasons: the existing tickets need to be re-evaluated due to changes in the company's situation, the company might be prioritizing the wrong things, or that the way of prioritizing tasks needs to be adjusted. To summarize, problems in the development process indicate whether the planned priority order is working or not.

In this case, the users want to see a dashboard view that resembles Trello. The view should be split in two parts: tickets that need action and a view for exploring various process metrics. The tickets that require action should be displayed accordingly to the problem's severity by size and color. Other visual indications can include opacity, borders, or filling of some sort. Furthermore, the view should be customizable for each user. The proposed customization options include color and threshold values for all problem indicators, and the presence of symbols or raw numbers. At a later time, each role should be able customize what information they see and how it is displayed.

The following entries all relate to tickets and provide concrete examples on indicating problems in the development process. The entries have been described from the product owner's perspective.

5.5.4. Demonstrator Specifications & Requirements

Demonstrator Title	Trello process management proof of concept
Author/company	Henri Bomström / University of Oulu
Partners involved	Invenco, University of Oulu
Primary Actors	The primary actor in this case is the product owner.

Requirement	Description	Priority (high/medium/low)
R-UC1.4-01 Ticket immovability	A ticket's immovability indicates that no one is doing anything about the ticket. The reason for visualizing ticket immovability is that the existing plan for ticket importance and urgency needs to be re-evaluated. Immovability can be measured by the time a ticket spends in the same state or swimlane, or by examining the time it spends in a certain quadrant of importance. Other indicators include the lack of views or opens for the ticket, or that no new information is added to the ticket.	High
R-UC1.4-02	Abnormal changes in ticket state indicate the need for re-evaluating the ticket priority order. The	High

Changes in state or swimlane	company described abnormal state changes as a measurement for the “windiness” of development. High windiness can indicate deeper problems in prioritising and planning tickets as a whole. Abnormal state changes can be detected if tickets skip steps from the normal development process or change lanes back and forth.	
R-UC1.4-03 Changes in priority order	Changes in ticket order indicate that the existing plan for ticket importance and urgency has changed. Normally tickets are presented and processed in priority order. However, sometimes urgent requests or changes in the company’s situation can change the order of tickets. When a ticket noticeably drops in priority order, the feature should be re-evaluated against the current situation. Changes in priority order can be measured by the ticket’s vertical position on the swimlane.	High
R-UC1.4-04 Board output	Board output indicates the average time a ticket spends on the board. By examining the board’s output, a product owner or a manager can better estimate how long certain development tasks take. This information can be used to produce more accurate work estimates. Board output can be measured by counting the average time in days tickets take to move through the board. If possible, the type and size of tickets should be taken into account when reporting board output.	Medium
R-UC1.4-05 Ticket reappearance	Ticket reappearance indicates that similar tickets have dropped out of scope multiple times. Reappearance can indicate that the feature is more important than previously thought of.	Low

6. USE CASE 2: ENTERING INTERNATIONAL SERVICE BUSINESS.

6.1. Introduction

Vincit is a software development company focused on developing embedded systems, web and mobile apps and service design. In this use case, Vincit aims to experiment in introducing its leadership as a service (LaaS) system into the European market. The LaaS system aids in organizational change management and Vincit aims to create a product family around it, where each service solves a certain problem. In order to realize this goal, Vincit aims to lower its lead times from development to production on delivering product features.

The current use case focuses on identifying usage problems in their LaaS product. The aim of identifying these problems is to determine whether there are subtle annoyances present in the UI or process flow of their product that doesn't emerge through regular client feedback. Other important measures include client dropout risks, user activity, organizational development towards LaaS and LaaS compatibility.

6.2. Actors & Stakeholders

The primary actors in Vincit's case are managers, employees and customers. Manager roles include software architects, client contact representatives, product owners and design executives. Employees can take the roles of developers, customer support and concept designers. From the customer's side, the client company can be thought of as the highest-level actor. Each customer company has regular users and nominates a LaaS team that is responsible for LaaS usage and will be trained by Vincit. Furthermore, each customer company has two special roles: super users and service providers. Super users receive more training from Vincit and support other users in the organization. Service providers are responsible for the services provided in the organization. Vincit offers special training for this role.

Customer representative managers act as communicators between the client and Vincit. User feedback may also be relied through these representatives. Development related managers guide feature releases and work prioritization.

6.3. Business Needs

Vincit seeks to better understand if LaaS philosophy is doing what it should be doing in a client company. In order to determine whether users are correctly using the LaaS system, additional measurement points are needed besides user activity. Vincit is interested in identifying whether there is something wrong with the Laas UI or process flow that won't emerge through regular client feedback.

Super users are the engine that propagates LaaS usage through the organization. However, these users can become problematic when the product is used in a different way that is intended. Long time "wrong" usage can result in users learning to use the system in a wrong way. Vincit wants to find these users but the visibility of user errors is lacking. Sometimes user errors can be caught from error logs but are hard to debug and won't necessarily convey the real problem. Identifying and reporting user anomalies is difficult as the data as not readily available. Even if user activity data could be collected, comparing user behaviour with current

tools is either hard or impossible. Better understanding of LaaS usage would help Vincit retain customers by minimizing subpar user experiences.

Direct customer feedback is important for development teams as all the relevant information cannot be found through the LaaS system. Currently, user feedback is processed in Trello. The problem with this approach is that Trello offers only one grouping possibility and might not scale with increased customer base. Users group tickets based on what is to be done about the feedback. However, the original thought behind each ticket can be lost as time progresses. Visibility of client feedback is important as it tells the clients that they are being listened to and something is being done about their feedback.

6.4. Users' Perspective

Feature roadmap is an important concern that would help make development efforts and priorities visible. Architects need to be able to look a year ahead in order to make informed decisions on structure and technology selections. Currently, the big picture of development pipelines can be hard to see as the visibility provided by the combination of Jira, Git and Gerrit can be too coarse. There should be a clear view of what should be done, when someone has started working and which features are ready for review. A clear view of development would aid in task prioritization and communicate feature weights to sales for more accurate description of development possibilities.

Sales on the other hand could benefit from seeing which features could be developed simultaneously. There is a need to identify when the company's needs align with client needs to synergize development efforts. In order to achieve this, the conversations between sales and clients should be somehow identified and visualized. Feature requests could be sorted by count and business volume to identify related development branches.

Customer representatives would like to see how LaaS is working in a client company. This can be achieved by representing problems with "traffic lights" to identify problem areas in customer organizations. The benefit of identifying customer problems is that representatives can better identify what kind of help the customer might need. Furthermore, customer feedback could be visualized in numbers by indicating how many tickets ended up on backlogs and releases. This would communicate that the client is being listened to.

6.5. Demonstrator Specifications & Requirements

Demonstrator Title	Visualizing LaaS usage
Author/company	Henri Bomström / University of Oulu
Partners involved	Vincit, University of Oulu
Primary Actors	The primary actor in this case is the product o

Requirement	Description	Priority (high/medium/low)
R-UC2-01 Key user activity	Activity of super users and service providers. Vincit should be able to see if super users are up to their tasks and use the system actively. The super user should see the service provider situation and what should be done about it.	High
R-UC2-02 User irritation	Identification of clients whose users grow especially irritated. Measured with an index that is constantly improved on. Visualized on a meter or a graph.	High
R-UC2-03 Client dropout	A measurement on how well the client utilizes LaaS. A client specific relative value visualized as a graph or a number that resembles stock market visuals. The aim is to address largest drops.	Medium
R-UC2-04 Organizational movement towards LaaS	A number between 0 and 100 that represents a trend on movement towards or away from LaaS. Supervisor time usage and self-direction can be used as meters on evaluating progress.	Medium
R-UC2-05 LaaS compatibility	Visualization on what is required for LaaS to work. Aims to facilitate calibration of LaaS in a company.	Low

7. USE CASE 3: VISUALIZATION IN TEACHING

7.1. Introduction

A typical situation in teaching experienced by VISDOM education partners is that students' problems in learning remain unknown until when an exam or an assignment is due, which is often too late to help a student to pass a course. This is highlighted in learning modes where there is little or no direct contact to students such as in MOOCs and distance learning and in cultures where losing one's face is an issue. Even in the cases where there are good contacts to students, the students may intentionally or unintentionally overly give positive impression on their situation. Modern educational environments, especially in IT technology, use both programming tools such as IDE, VCS, and automated testing in the evaluation of assignments and MOOCs. This opens new opportunities to provide visualizations to teachers in order to get an up-to-date understanding on students' progress and also to grasp learning problems in the early phase. The use of visualisations also enables MOOCs and distance learning and hence, and also to selling education to remote cities and countries. Finally, visualisations can be used in SWE project courses by project groups in the same way as in SW companies. This way students get exposed to modern tools during their studies and later, when the students get employed, they will spread the knowledge on the tools among SW industry. The four universities of the consortium will collaborate in the creation of demonstrator that shows use of VISDOM dashboard in teaching.

7.2. Actors & Stakeholders

The main actors of this use case are teachers and students, but in some cases also support and administrative people. The standpoint of each use case is shown below:

Teachers (this is still an initial list, more detailed analysis will be based on conducted interviews)

- Can track learning progress of students (as a whole and as a group)
- Recognize the students that would need special attention
- Continuously improve the course.

Students:

- Can get feedback on their progress
- Can plan the activities
- Can get more insightful learning

Support people who help students in their study plans

- Can get insight of the background of the students
- Can understand the course

Admin people

- Can have a better view to the statistical data

7.3. Business Needs

Increase and improve the use of MOOC's and on-line learning. Use of on-line tools in teaching is an increasing trend. These tools are useful for teaching the new professionals in the universities and especially when the skills of people already working need to be updated. Furthermore, the business of exporting education is increasing.

Improve the learning results. the Visualization of the development work in learning phase would help teachers and students in getting insight and feedback of progress. Thus, it will have a big impact on performance of the education, teaching export and life-long learning.

Improve the efficiency of using resources of teaching staff. With the help of visualization, the teacher can track well-progressing students with a minimal effort, and also to concentrate their attention to students that need help.

7.4. Users' Perspective

As the visualizations are based on data, privacy concerns need to be taken seriously. The different stakeholders should have access to different subsets of the data at different times,

Different stakeholders have different information needs and different backgrounds. This means that dashboards should be configured to differently for different stakeholders. Also, the visual notations should be familiar and suitable for the backgrounds of the stakeholders.

7.5. Demonstrator Specifications & Requirements

Demonstrator Title	Visualization is teaching (of software engineering)
Author/company	Kari Systä / TAU
Partners involved	TAU, U.Oulu
Primary Actors	Teacher; Student

Requirement	Description	Priority (high/medium/low)
R-UC3-01 Data	The teaching demonstrator will be based data from a real course, but the data is carefully anonymized.	High
R-UC3-02 Stakeholder	Teacher is the primary stakeholder and the demo shall inspire teachers to give feedback and new ideas.	High

R-UC3-03 Visu 1	Show student activity over time to see if student have distributed they time evenly over time (maybe a new pulse-visualization)	High
R-UC3-04 Visu 2	Show worksplit among the team to see that everybody participates (simple charts, may some innovative EKG visualization)	High
R-UC3-05 Visu 3	Show derived visualizations to put attention to students that may need extra guidance. (x-ray)	High
R-UC3-06 VISDOM	Demonstrate the idea of as many visualization type (EKG, Pulse, blood pressure and X-ray) as possible.	Medium
R-UC3-07 VISDOM	The visualizations need to run in a configurable dashboard.	Medium