NTNU
Norwegian University of
Science and Technology

Supporting Agile Processes within the Norwegian Infrastructure Industry

Integrating BIM-software with task and process management tools

Øystein Bjerke Hauan

Master of Science in Engineering and ICT
Submission date: May 2018
Supervisor: Bjørn Andersen, MTP

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Preface

Through my supervisor at NTNU, Professor Bjørn Andersen, I was introduced to Jan Erik Hoel, an employee at Trimble Solutions Sandvika. He was a part of a Norwegian R&D project with the objective of reducing the planning and design time of infrastructure projects by up to 50%. I had a desire to develop and program an application in my master's thesis, and we agreed to collaborate on a project thesis in the fall of 2017. Following the success of that project, we defined a master's project in collaboration with Bjørn, and this thesis is the result of this collaboration. It concludes my five year integrated master's degree in Engineering and ICT - Project and Quality Management at NTNU.

Many people have been involved in and made contributions to this project, and for this I would like to extend my sincere gratitude. There are a few of these individuals I would like to mention specifically.

First of all, I would like to thank my supervisor at NTNU, Bjørn Andersen, for his help in designing this thesis, and providing valuable feedback throughout the project.

Secondly, I would like to thank my supervisor at Trimble, Jan Erik Hoel, for making this project a possibility. He has done a fantastic job of introducing me to a large number of experts in the industry, and pushing the project forward, while providing me with all the resources I needed.

Thirdly, a large part of developing the IssueConnector has revolved around designing it's architecture. Anders Hauan, my father who just so happens to be a seasoned software architect, has assisted me with this throughout the entire project, while showing great interest along the way, and also providing insights and good practices.

I would also like to thank the following persons for their contributions:

- Pieter Dhondt, head of web development at Trimble Sandvika, for his assistance during development.
- Steinar G. Rasmussen, for his contributions throughout the project, and also for his thesis from 2016, which has been an invaluable resource to this project.
- Ingrid Hauan and Joakim Frimann-Dahl, my sister and brother-in-law, for proofreading and providing tips on how to write and structure this thesis.
- The representatives from Sweco, Rambøll, Cowi and Hæhre, who have shown great interest in the product, thereby providing great motivation.
- All the participants in the SPP project, for their motivation and interest in the project, as well as insight in the industry.

I consider myself privileged to have been given the chance to do this, and I am overwhelmed by all the positive feedback I have received during the project.

Summary

The Norwegian infrastructure industry is suffering from long lead times for large projects. This is due to several factors; the fragmented nature of the industry, the lack of involvement from stakeholders and project owners, as well as the lack of transparency and involvement between disciplines, to name a few. The Norwegian government has established an objective to reduce the planning and design time for these projects by 50%. There are ongoing initiatives for implementing Concurrent Engineering in the industry, a process methodology aiming to execute the different phases of a project simultaneously, as opposed to sequentially.

Samtidig Plan og Prosjektering (SPP) is an ongoing Norwegian R&D project, with the main objective of exploring new process methodologies and technologies to achieve the aforementioned time reduction objective. Experts from several companies in the industry participate in this project. One of the sub-tasks is to explore ICT tools that can aid in reducing the planning and design time, which is what this thesis concerns. By drawing inspiration from agile software development, this project explores how tools used for software development projects (in this case Jira) can be configured to be suitable for the infrastructure industry, and analyzing the potential of such tools based on pilot projects and user feedback.

In addition to this, the thesis describes how an integration platform was developed, with the objective of connecting various systems in the infrastructure industry, named the IssueConnector. The IssueConnector connects systems such as BIM-software to the aforementioned tools for software development projects. The purpose of this connection is to eliminate the need to register updates in more than one system, by executing automatic updates in all connected systems in real time. It was developed and tested entirely during the course of this project, and it managed to gain a lot of interest in the infrastructure industry. An indefinite amount of systems is able to connect to the IssueConnector, but for this project, it connected BIM-software (NovaPoint/EasyAccess) to Jira, an issue tracker for software development projects.

The project resulted in two successful pilot projects for Jira and the IssueConnector, connecting Jira to EasyAccess and synchronizing these in real time. Both pilots were conducted by companies on real infrastructure projects, and the feedback from both pilots were unanimously positive. It is broadly agreed upon that solutions such as the IssueConnector have great potential in the industry.

A questionnaire regarding the potential of Jira was also distributed to experts in different disciplines within the infrastructure industry, to people who had already tested Jira. The result of this states that almost all of the respondents believe Jira has potential in the industry, and that it will greatly contribute to the overall project management of such fragmented projects.

Table of Contents

List of Figures

Abbreviations

API	=	Application Programming Interface
BCF	=	Open BIM Collaboration Format
BIM	=	Building Information Model
CC	=	Concurrent Construction
CE	=	Concurrent Engineering
CRUD	=	Create, Read, Update, Delete
EA	=	EasyAccess
FRx	=	Functional Requirement X
GUI	=	Graphical User Interface
GUID	=	Global Unique Id
IC	=	IssueConnector
ICE	=	Integrated Concurrent Engineering
IFC	=	Industry Foundation Classes
KPI	=	Key Performance Index
NFRx	=	Non-functional Requirement X
RAM	=	Reliability, Availability and Maintainability
REST	=	Representational State Transfer
SOAP	=	Simple Object Access Protocol
SPOF	=	Single Point of Failure
SPP	=	Samtidig Plan og Prosjektering
UI	=	User Interface
VNPT	=	ViaNova Plan og Trafikk

Chapter 1

Introduction

1.1 Background

The Norwegian government defined an objective in the National Transport Plan (NTP) 2014-2023 to reduce the planning time in large infrastructure projects by up to 50% [3]. The process of planning these large projects is highly complicated and fragmented, partly due to the number of different disciplines involved. Every participating group has a separate involvement in the process, leading to long lead times and confusing communication. Important project decisions are typically made at meetings, regular or irregular, which can often result in long delays when other work is forced to wait for these decisions. As the communication between disciplines will primarily be over the phone or e-mail, there is a high risk of misunderstanding and miscommunication, with low levels of transparency between participating groups throughout the project.

The different processes, or phases, have traditionally been executed more or less sequentially, meaning that one phase of the project will wait until the previous is entirely finished, before being able to start. This naturally leads to long lead times, and also great risks of major delays, which is not uncommon for Norwegian infrastructure projects. [4, 5] Therefore, efforts are being made towards increasing the efficiency of these processes. One of the approaches being tested is implementing Concurrent Engineering (CE) methods for parallelizing the different phases and processes to decrease the amount of time from project start to it's completion. This method is already being used successfully in other industries; one central example for Norway would be the oil and gas industry. [6]

This thesis has been written in collaboration with an ongoing R&D project, called *SPP* (Samtidig Plan og Prosjektering, or Concurrent Planning and Engineering), with the purpose of fulfilling the government's time reduction objective. There are several different components to this project, aiming to improve almost every aspect of the current situation. One of the sub-objectives of the SPP-project is exploring the possibilities of using or developing software to aid the process of designing and planning infrastructure projects. As the author's field of study is a crossover between software development and project management, it is primarily this aspect of the project this thesis concerns. It is preceded by a project thesis, which explored different tools for project management, primarily from software development. The latter created a basis for this final thesis, where the main objective ultimately has consisted of two parts:

1. Configure previously explored software development tools to enable usage in the Norwegian infrastructure industry.

2. Develop an integration platform prototype, to enable seamless communication between established, industry related software, and project management software.

1.2 The Norwegian Infrastructure Industry

Projects in the Norwegian infrastructure industry are affected by long planning and design processes, which naturally lead to undesirably long lead times. As a response to this issue, the Norwegian government established the aforementioned time reduction objective. The potential for reducing the lead time for these types of projects is significant, and the utilization of modern technology (issue tracking and project management software etc.) in order to achieve this is long overdue.

One of the main areas with great opportunities for improvement is the introduction of multidisciplinary, simultaneous cooperation in the early phases of a project. Decisions made early in the projects tend to have little weight in later phases, and it is often unclear why the decision was made in the first place [3]. This can lead to decisions made by an expert in a certain discipline, being overturned or even made *again* in a later phase, due to the low level of involvement and transparency between the participating disciplines.

There are ongoing intiatives to reorganize the official planning process in Norwegian projects (Plan og Bygningsloven)[7], as well as for improving the design and construction process of these. One of the largest initiatives established by the government, was eliminating Statens Vegvesen (Norwegian Public Road Administration) as the sole contractor of Norwegian road projects, and introducing a second, state owned corporation. This is called *Nye Veier*[1] (New Roads) and it is responsible for planning, building, running and maintainance of certain large roads in Norway. The establishment of Nye Veier represents a major shift in the industry, in terms of contract regimes and requirements that are set for contractors and engineers. Earlier involvement of the contractors in the planning and design phases and the elimination of technical drawings to focus on BIM-models, are a few of the new requirements that disrupts the entire infrastructure industry.

[1] `https://www.nyeveier.no/`

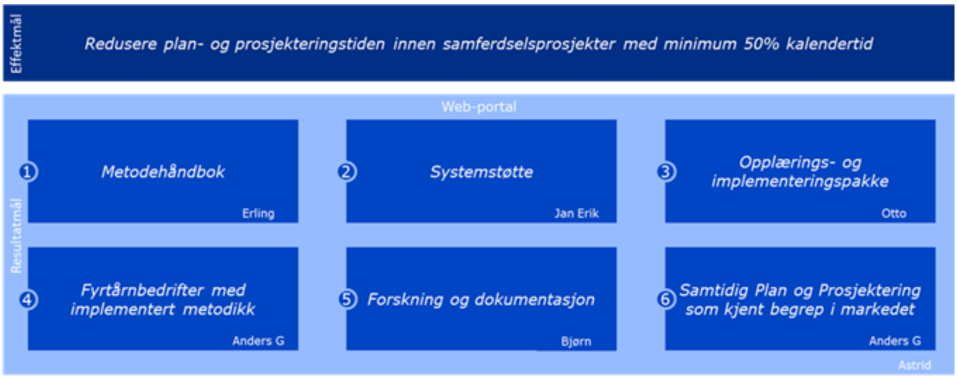


Figure 1.1: An overview of the objectives of SPP

1.3 The SPP Project

The SPP project consists of participants from several companies in the infrastructure industry, including Trimble, ViaNova Plan og Trafikk, Sweco, Rambøll and Metier, among others. The objective of the project is to reduce the planning and design time for large infrastructure projects by 50%. The timespan for Norwegian infrastructure projects average around ten years, according to Jan Erik Hoel in Appendix E.

The project has several intermediate objectives, among others (as found in Appendices C and E):

- To research and evaluate existing, relevant practices, other known methods and measuring KPI's (Key Performance Index) of these methods.
- To develop test- and pilot-projects, and to review two Integrated Concurrent Engineering (ICE) projects at Bane Nor.
- To collect knowledge and best practises of how to organize concurrent sessions and other processes that support the concurrent sessions
- To develop a digital handbook (web pages) for best practices
- To develop software solutions to support the processes
- To test the researched processes in real projects
- To inform the industry about the findings through workshops, seminars, articles etc

An overview of the SPP objectives, made by Otto Bergersen at Metier, is displayed in figure 1.1.

One of the expected outcomes of these objectives is to find existing or develop new tools to support planning, engineering and decision-making. In other words, to support the main administrative processes of a project. This is part of what this thesis explores.

1.4 Problem Statement

The problem statement for this thesis is closely tied with the aforementioned goals, thus is it split in two. They are as follows:

1. P1: Can tools designed for managing agile software development projects (in this case Jira[2]) be utilized in the infrastructure industry, and what amount of configuration/tweaking is necessary to make it an equally powerful tool?
2. P2: Develop a platform that seamlessly combines already established and well known industry related systems (in this case NovaPoint and Quadri [3]) with new project management software (Jira). The function of this software is to remove the need for making the same changes in several systems, and to ease the process of using multiple digital platforms for the average end user.

1.5 Jira

Jira is the selected tool for exploring P1. It is an issue tracker designed for software engineering projects, allowing for extensive control and monitoring of projects, as well as estimation and reporting. It was selected on a basis of acknowledged requirements, described in chapter 4.2, as well as a previously conducted technology analysis (Appendix A).

1.6 IssueConnector

The IssueConnector (IC) is a prototype software, developed entirely during this project, to satisfy P2. It is a solution that enables live connection and integration between modern software tools. The IC is designed in a way that allows for an indefinite number of systems to connect to it, without modifying existing connections. When two or more systems are connected, they share a common data format, allowing updates made in one system to be registered in the connected systems. This all happens in real time, without delay. It is a *backend* system, meaning that no user interaction is directly involved in the software, except for administrators.

A demo video of the IssueConnector is available here[4]. The source code is available here[5].

1.7 Structure

This thesis consists of six chapters, and twelve appendices. When developing new software solutions, it is important to document these thoroughly, both to enable continued development by other parties, and to allow utilization of the solutions.

[2] `https://www.atlassian.com/software/jira`
[3] `https://www.novapoint.com/home`
[4] `https://youtu.be/wKvlV9b47gc`
[5] `https://github.com/oysteinhauan/IssueConnector`

Although not all of the documentation mentioned is explicitly relevant for the object of this thesis, it is included to offer insight into the general field of study.

Chapter 1 - Introduction
This chapter gives an overall introduction to the thesis, describing the goals of this project as well as the larger project this is a part of. The problem statement is presented in this chapter.

Chapter 2 - Method
This chapter describes the process of conducting this project. How decisions were made, how the software was designed, how the meetings were conducted etc. It also describes how relevant literature was selected.

Chapter 3 - Theory
This chapter presents some relevant theory for this project, and explains the need for change in the industry. The theory presented is based on previously conducted studies and papers.

Chapter 4 - Jira
In this chapter, Jira is presented as a project management software for infrastructure projects. It includes a description of the configurations, future possibilities, feedback from previous test projects and current projects.

Chapter 5 - IssueConnector
This chapter describes the goal of the issue connector, the development process, and it's architecture, as well as solutions to various obstacles faced during development. It contains how the messages are handled, as well as a thorough description of the various layers of the software. Additionally, there is an analysis of user feedback.

Chapter 6 - Conclusions and Future Work
This development project had a very short duration of only 5 months, which means that a lot of work, especially on the IssueConnector, remains before it can be released as a functioning solution. This final chapter summarizes the solutions to the problem statements, future possibilities, potential impact, and future challenges.

Chapter 2

Method

The project explores the possibilities for using modern software development tools in infrastructure projects, and the development of a connection between already established software used in the industry and new tools for task management. This chapter describes in detail how each of the problem statements were solved, as well as how the theory chapter was written.

The project has consisted of several research methods. The most significant being the interviews and meetings with experts in relevant fields. This has helped in distinguishing the problems that require solutions. A participant in these meetings is the author of "*Integrated Concurrent Engineering i Samferdselsprosjekter*" [3], which is a paper serving as a foundation for this project. He has also been active in the requirement specification and testing of the IssueConnector, and developing a prototype Jira configuration.

2.1 Theory

The available research literature regarding concurrent engineering in the infrastructure industry is limited, making this project exploratory to some extent. The theory chapter is based on available literature on CE in the oil and construction industry, as well as on agile software development and challenges in the infrastructure industry, as observed and described by participants in aforementioned meetings. The objective of the chapter is to justify the need for software such as Jira and the IssueConnector in the infrastructure industry.

The literature used for this chapter has been found using the following search queries:

- Q1: (concurrent engineering) AND infrastructure
- Q2: (concurrent engineering) AND construction
- Q3: concurrent AND engineering AND construction
- Q4: (concurrent engineering) AND oil

- Q5: concurrent AND engineering AND oil
- Q6: (agile development) OR scrum

The queries were executed in Bibsys, IEEEXplore, Google Scholar, ACM Digital Library, ScienceDirect and Scopus. The results of these queries were then screened using a set of predefined criteria, based on the title and abstract of the result. Below are the inclusion and exclusion criteria for the results regarding concurrent engineering:

Inclusion Criteria	**Exclusion Criteria**
Paper is written after 1980	Study is written before 1980
Discusses CE in construction or infrastructure or oil and gas	Mainly discusses CE in manufacturing

Finding literature on agile software development requires less effort, as this is a method that has been researched, documented and evaluated for several years. The main sources here have been textbooks from previous courses on the subject, attended by the author at NTNU.

Relevant findings from the articles were then extracted, and presented in this chapter to form the theoretical foundation of this thesis.

2.2 Configuring and Testing Jira

The first main objective of this thesis was to configure Jira in a way to make it suitable for use in non-software related projects, specifically in Norwegian infrastructure projects. As the author has a software related background, it was important to ensure involvement from end users in the targeted disciplines to determine a suitable configuration for their needs. Therefore, the involvement of different disciplines from various companies proved beneficial. The most prominent of these were Sweco, Rambøll, Hæhre Betonmast and Cowi, among others. Sweco and Rambøll have, prior to this project, tested Jira as an alternative for managing their infrastructure projects.

The first challenge was determining exactly what one wishes to achieve with this software, and why the problem with long lead times is present today. In other industries, such as the oil industry, this problem is not nearly as evident. Steinar Rasmussen explores this in his thesis from 2016, which has been used as a guide to understanding the essence of the problem. He states that a lot of the problems arise from the low level of involvement between disciplines and stakeholders. The infrastructure industry is disciplinary fragmented to such an extent, as making the individual work of each discipline more or less isolated from other disciplines [3]. Ultimately, one of the most attractive features Jira provides is the ability to observe all tasks at any point in the project, with out being directly assigned to it. The idea is that this will lead to an increased level of involvement between disciplines.

Initially, the decision log model in Appendix B (by Erling Graarud) was used to to create a "copy" of this in the Jira software. The plan was to use this foundation as a test project for the IssueConnector software that would be developed at a later stage. Interestingly, the IssueConnector immediately gathered an overwhelming amount of attention from the industry once partially developed, with all the aforementioned companies wanting to test it for their "real-world" projects. This lead to the development of the IssueConnector being prioritized over the further configuration based on appendix B. Some of the interested companies, like Sweco and Rambøll, had their own Jira installations, and they wanted to connect it to NovaPoint/EasyAccess[1] using the IssueConnector. (EasyAccess is a web solution connected to NovaPoint, allowing to define topics and share these in across devices). Hæhre, on the other hand, did not have Jira, so it was agreed that one would installed for them, and accordingly configured together.

Through workshops, a list of requirements for their Jira was generated, as displayed in Appendix D. Both Sweco and Rambøll, who had already tested Jira, offered their configuration as a source of inspiration.

In exchange for being allowed to use the IssueConnector, the involved companies agreed to provide feedback for evaluating the potential of both Jira and the IssueConnector. This was performed through a series of enquiries, both in the form of a questionnaire, and via e-mail correspondence. The results from these questions are available in appendices F and H. These were then analyzed and discussed in chapter 4.

2.3 Developing the IssueConnector

2.3.1 Architectural decisions

A significant aim of this project was to design a fitting architecture for the IssueConnector. A few pre-requisites exist in order to develop a suitable design. First of all, it must be clear *what exactly* is to be developed; What ought to be its functions, who will be using it, what problem is it solving, how will the problem be solved, and so forth.

Requirements of the software should also be defined, to some extent. When designing an architecture, the most important requirements to consider, are usually the non-functional requirements. The difference between functional and non-functional requirements is essentially that functional requirements describe *what* the software is going to do, and non-functional requirements specify *how* it should execute its tasks. For example, a functional requirement for a search engine would typically be that you should be able search by using text input. A non-functional requirement could be how fast the search should terminate.

The requirements for this software were defined through various meetings during initial development, and evolved along with the development process. The participants in these meetings usually consisted of the project supervisor at Trimble, Jan Erik Hoel, and Pieter Dhondt, Web and Mobile Solutions Architect at Trimble

[1] https://www.quadridcm.com/EasyAccess/Intro

Sandvika, in addition to the author. It should be noted that these meetings were strictly software related, which means there was not much need for input from people in the infrastructure industry to define these requirements. The requirements were limited to the scope of this project, to ensure that completion within the very short time limit of this project was feasible.

There are potentially endless possibilities when it comes to choosing an architecture for a system. There are many factors to consider, and there is usually no definitive answer to what the optimal solution is. This project focused on the most important requirements, namely that the system should be sending messages between an indefinite number of systems, and that it should be possible to add new services to the system without impacting any of the existing services. For inspiration and guidance in the choice of path, Anders Hauan offered insights based on his more than 30 years of experience in software architecture. He was presented with an extensive explanation of how the system ideally should work, using the then developed prototype to further portray the actions of the IssueConnector. After ongoing discussions a suitable patern for this application was agreed upon, further discussed in chapter 5.

Hauan has during the entirety of this project functioned as a counselor and expert on both architectural and programming decisions.

2.3.2 Development and Test Projects

The IssueConnector is developed entirely by the author, using an iterative approach. Initially, it was demo-oriented and "rushed", before it was decided that the application should be rewritten based on a reworked architecture and new design patterns. The development has followed agile principles, with technical meetings every week, where progress and challenges were discussed.

After only a few weeks of development, we launched test projects, both internally and externally. The testers were allowed to try the software, on the premise that they would provide feedback and data for this thesis, based on their experience and expertise. The analysis of the IssueConnector's potential is based on data collected from these tests.

2.4 Meetings and Sessions

Over the course of this project, multiple meetings of various topics have been held, some of which were held in direct consequence of this project. These meetings have enabled the interaction with experts from many different fields, and given an opportunity to explain what this project concerns. The IssueConnector was usually demonstrated in these meetings, thus it gathered attention. As stated in [1], the industry needs a redefined standard for supporting advanced task management across different tools.

A knowledge transfer meeting between SpeedUp[2] and the SPP project, defined an inception of the IssueConnector. Attending this meeting, hosted by Sweco

[2] https://www.sintef.no/prosjekter/speedup-kortere-gjennomforingstid-i-prosjekter/

Vækerø, were around 40 participants from different backgrounds in the industry. Towards the end of the meeting, a short presentation was held on the IssueConnector, followed by a quick demonstration of the first prototype. After this presentation, several people wanted to sign up to test the application, , and this defines the first time the author realized the widespread potential of the IssueConnector. It was clear that focusing on application stability became very important at this point, if people were going to start using it. That sparked the architecture refactoring described in chapter 5.2.

The SPP project hosted several meetings that I attended. In spite of the meetings not always being directly relevant to this project, the SPP project is the foundation of this project, and its starting point.

Chapter 3

Theory

The ideas behind the agile software development methods can be traced back to the 1970's, although the Manifesto for Agile Software Development was formulated in 2001 by a group of seventeen software developers, and this is what marks the beginning of what is referred to as modern agile software development [8]. The impact of this method on software development has been considerable, and proves interesting because of the way it addresses some significant drawbacks of common projects, e.g. cost calculation, concurrency or decision making. A team developing software using agile techniques will focus more on the continuous development of the product, paying close attention to customer requirements, and less on the overall documentation and process of the objective at hand [9]. The methodology opens up for constant re-evaluation of implemented solutions even late in the development process, and unlike several of the classic methodologies it has a much larger focus on continuous quality assurance and testing. The decisions made during the life cycle of a project is often made at developer level, reducing time for bureaucracy and waiting for decisions to be made by people who may not even be the most qualified to make said decision.

Concurrent engineering (CE), is a process or project methodology. It originated in the manufacturing industry, as a way to cut costs and time for designing and manufacturing products. The way it is achieved, is by considering all the downstream processes and requirements in the design stage [10]. This includes extending the involvement between disciplines and responsibilities, in order to gain a better overview of the entire process from the very beginning. In other words, it seeks to minimize the timescale by maximizing the overlap between phases and processes. The same methodology is transferred to other areas than manufacturing, as many types of projects would benefit from this shift from traditional, sequential execution to a more parallel, streamlined process.

An objective in this project is to explore what the infrastructure industry can learn from agile software development methods, to increase the efficiency and transparency of large projects.

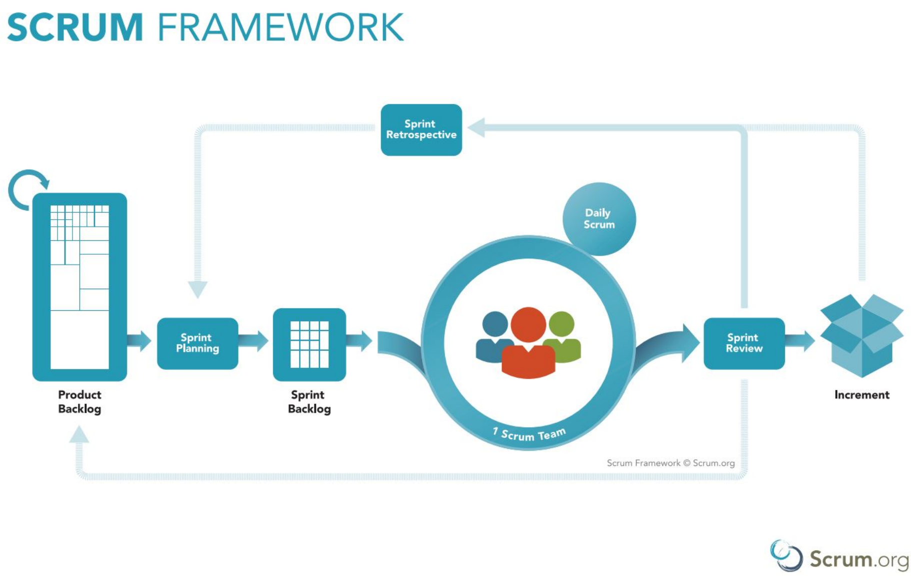


Figure 3.1: The SCRUM Framework

3.1 Agile and Scrum

Software engineering differs from other engineering disciplines, such as construction and infrastructure. An agile software development team will typically consist of a Team/Project lead, or Scrum master, who should both be able to successfully lead a group of developers, as well as having a basic understanding of relevant technology [8]. The team members will normally be comprised of back-end and front-end developers, perhaps some full-stack developers, QA/Test engineers, UX designers, and they are responsible for the documentation of their product. In addition to these, the stakeholders have various levels of involvement in the development(end-users, investors etc), and most products will have a product owner, who may or may not be involved in the day-to-day development. Despite consisting of several fields of expertise in these teams, the vast majority fall within the field of software engineering and computer science. The most widely adopted process framework in agile development is Scrum. [11]

"Scrum is a process framework that has been used to manage work on complex products since the early 1990s." [12]. The process is iterative and incremental, which is standard for agile processes. Actions or tasks, selected from the *backlog*, are divided into so called *sprints*, which is a fixed duration of time during which these tasks should be completed. These sprints are planned in sprint planning meetings, held at the very beginning of each sprint. It is also customary to hold *stand-ups*, which are very short meetings where each scrum member typically explains what they have done the past workday, and what they will do the following day. These meetings should be kept short, and preferably not exceed 15 minutes. This way, the team and its leaders constantly reevaluate the project progression. In addition, the whole team is informed on the status of the entire project, not just their individual tasks.

Scrum is designed for small teams in fast-paced development projects, typically three to nine members in total [13]. This is far less than what a typical infrastructure project will consist of, in addition to the latter being a lot more "scattered", with people from different fields working from different places. However, the industry has sought inspiration from this process, and tried to adjust it to fit with their needs. Figure 3.1 displays the iterative scrum process.

3.2 Concurrent Engineering

The definitions for CE are plentiful, and the following explains it well:

"*Concurrent Engineering (CE) is a systematic approach to integrated product development that emphasizes the response to customer expectations. It embodies team values of cooperation, trust and sharing in such a manner that decision making is by consensus, involving all perspectives in parallel, from the beginning of the product life cycle*" [14].

Through better integration of activities and processes, CE aims to reduce the life cycle time span of a product development [15]. In the manufacturing industry, this has been very important because of the obvious benefits of short time-to-market. This does not translate directly to the construction or infrastructure industries however, as the contractor will not have the same significant benefits of finishing a project sooner than the contract states [6]. CE has, however, been successfully implemented in the Norwegian oil and gas industry.

3.2.1 Concurrent Engineering in the Oil and Gas Industry

Oil and gas projects are large scale and highly complex. They require knowledge from several fields, including geologists, civil engineers, economists etc. [16] CE focuses on the integration of people from different fields and the cooperation between them. This integration is done, as previously described, in the early design phases of the projects. The involved members are brought together in large rooms to cooperate simultaneously, allowing the different disciplines to receive quick knowledge and feedback from other disciplines, a method often referred to as *Big-room* sessions [3]. Using this method early on in the project, reduces the amount of rework needed downstream, and ultimately saves both time and cost.

3.2.2 Concurrent Engineering in the Construction and Infrastructure Industries

A major difference between the oil and gas industry, and the construction and infrastructure industries, is the involvement of the project owners. While an oil and gas project is typically funded by owners in the same industry, construction and infrastructure project owners will usually select the best bid on a contract, and leave the rest of the project to the contractor. This makes the integration between stakeholders and contractors more difficult than in the oil industry [16].

The infrastructure industry also suffers from administration delays, and long, complicated bureaucracy processes, leading to more obstacles for implementing CE. Some of the main obstacles identified in the research paper "*The Application of Concurrent Engineering in Infrastructure Projects*" [17] are lack of integration, low level of autonomy, and insufficient preparation and practice in the methodology. The lack of involvement from external stakeholders is especially difficult for this industry.

3.2.3 Integrated Concurrent Engineering

In the thesis "Integrated Concurrent Engineering i Samferdselsprosjekter" [3], the author, Steinar Rasmussen, is researching the effects of Integrated Concurrent Engineering (ICE) in the Norwegian infrastructure industry. It is a method developed to increase cooperation between different disciplines in large-scale projects. Rasmussen states that today, the industry consists of people focusing solely on their own part in the project, failing to see the project as a whole. In order to deal with this, it is important to do something about the involvement and integration between participating disciplines and experts. [3]

Rasmussen defined ICE as follows (translated from Norwegian):

"*Integrated Concurrent Engineering - where tasks are solved in parallel in collaboration-sessions. Project participants, stakeholders and contracting entities design solutions and make collaborative decisions.*"

Using the possibilities that come with modern technology, the aforementioned Big-rooms can be designed to facilitate the needs of the involved participants in the sessions. One of the walls will typically be a multi-media wall, with large screens that can display relevant information. The participants are seated around the room with their own workstations. Decisions for the project are then made in collaboration, and the communication between participants is effective because they can easily interact.

The frequency of these sessions may vary, but a major part of successful execution lies in planning these sessions. There are different ways to do this, but a detailed plan should be drawn up before the sessions, detailing the work of each participant and the expected outcome.

Benefits of this method include [3]:

- The design phase lead time is reduced
- The disciplines are involved earlier in the project, allowing them to voice their needs and concerns. This reduces the need for misinterpratation in e-mail correspondence, or lack of communication in general.
- Participants gain a multi-disciplinary understanding of the project.

One of the challenges of using this modern approach, is convincing participants to adopt new technologies and processes that they are unfamiliar with, and refraining from methods they are accustomed to use. Achieving this is important for the successful implementation of ICE. This should be taken into account in this thesis, as chapter 4 describes a possible introduction of a new task management tool. Simultaneously, one of the main objectives of the IssueConnector described in chapter 5, is to ease this transition by connecting and synchronizing relevant technologies and thereby reducing the number of systems that changes need to be registered in. Hopefully, this will lead participants to be more motivated for a potentially disruptive change in their workflow.

3.3 Digital Collaboration Standards in the Infrastructure Industry

There are a few key terms regarding digital collaboration in the construction and infrastructure industry. BIM and BCF are widely used in Norway, and it enables sharing and exporting object models and tasks between people and software.

3.3.1 Building Information Modeling

The US National Bim Standard defines BIM as:

"[...] a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition." [18]

BIM is widely adopted in the construction and infrastructure industries all over the world. The BIM object has several features, among these are:

- Information content that defines a building
- A geometrical representation of the building
- Visualization data, enabling a digital rendering of the building

[19]

The BIM forms a reliable basis for decisions made during the life cycle of a project. In a report from McKinsey[20], it is stated that 75% of companies that adopted BIM as standard procedure reported positive return on investment. [21]

3.3.2 IFC and BCF

Construction and infrastructure projects tend to consist of participants from several disciplines. These participants may use different software to work on the model at hand. To support the sharing of model data between these participants, an open standard called Industry Foundation Classes (IFC) was introduced by buildingSMART, an international organization aiming to improve information exchange between software applications in the construction industry [22, 23]. This standard is widely adopted in the Norwegian infrastructure industry [24].

Expanding on this is the BIM Collaboration Format (BCF), which is the format used to transmit issues connected to the IFC models [25]. The connected issues are called *topics*, and contain fields for author, assignee, creation date etc., in addition to comments. Consequently, project participants can exchange findings and issues in addition to the models. The problem with BCF is that it's insufficient in the face of more complex task management [1]. The authors of [1] suggest expanding the BCF format to "support the exchange of information for a broader range of process activities". The shortcomings of the BCF are the reason this standard has not been used for this project, as discussed in chapter 5.

If the BCF is expanded, it would be a natural choice to use as a standard in the IssueConnector. It could then be able to handle more complex task, issue and workflow management. An architecture for an expanded BCF is suggested in figure 3.2. This is based on a layered architecture pattern, i.e. each layer in the architecture is dependent on the one directly below. This way, the BCF format can be expanded and utilized with a varying degree of complexity, and tailored to fit the needs of the user. Using the fully expanded BCF format could enable the user to connect information such as risk- and cost management directly to the BIM-model. [1]

3.3.3 Shortcomings in BCF for the IssueConnector

The BCF format allows for exportation of BIM models with accompanying information as explained above. But with the IssueConnector, there is more attention paid to tasks accompanying the models. The IssueConnector will function as an integration system between different software, and will continuously update connected systems when an update is registered in one of them. When a comment is registered in one system, it should immediately be added to the connected systems where comments are enabled. The BCF does not support this, as it as designed for bulk export. This also applies to updates on status and priority, as the Issue-Connector will then update *just* those fields. Defining a new standard for the IC became the simplest solution for this project.

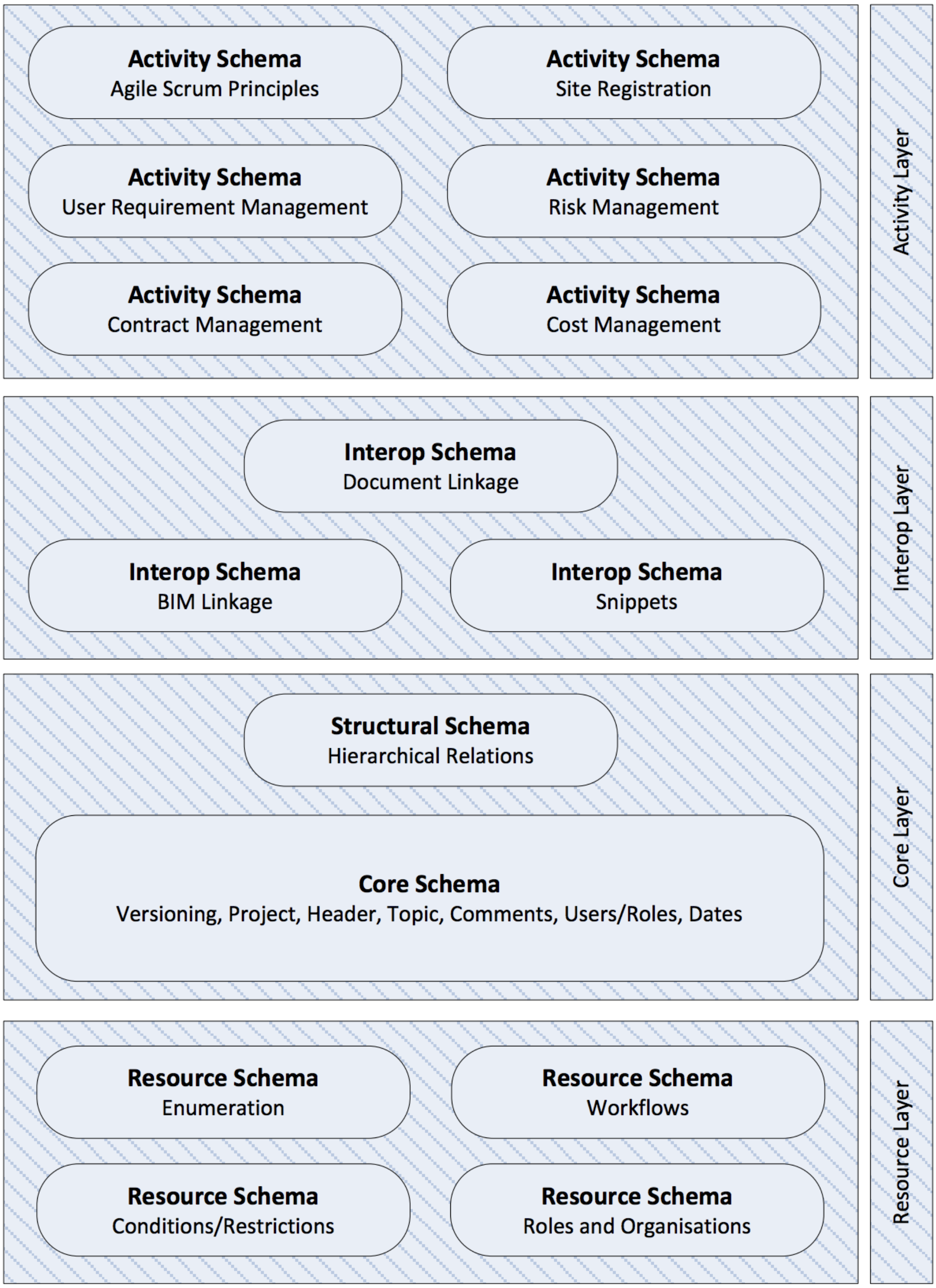


Figure 3.2: Suggested architecture for BCF expansion [1]

Chapter 4

Jira

The agile process is focused on delivering quality products quickly, without the same requirements to extensive documentation, as described in chapter 3. To support this process, it is common to use different tools. The most important are version control (e.g. Git), issue tracking and task planners. There are seemingly endless possibilities when it comes to selecting tools, and one of the most popular issue trackers is called Jira[1].

4.1 An Introduction to Jira

Jira is a project management tool designed for agile teams, primarily for software development projects. It is owned and developed by *Atlassian* [2], a company that owns many popular and widely praised development tools, including *Trello* [3], *Confluence*[4] and *BitBucket*[5]. It is a very powerful tool, allowing users to create so called *issues*. These issues can be divided into sprints and kanban boards, and they are highly configurable regarding what information an issue should contain. They can be assigned to people or groups, be prioritized, have an issue-specific workflow, to name a few.

Issues have different types, and for a typical software development project, the types would include *Bug*, *Epic*, *Feature* etc. These issues can then be tracked by updating the workflow, or status, of the issues and placing them on so called *Boards*, typically resembling Kanban and Scrum boards (see figure 4.1). These boards present a visual display of the project, where each board can be configured to display only a selection of the issues, based on types, due date, assignees or whatever requirements one may have. The ability to "tag" other project participants is an important feature, as it allows for extensive control of who is in charge or

[1]www.atlassian.com/Jira
[2]www.atlassian.com
[3]www.trello.com
[4]www.atlassian.com/confluence
[5]www.bitbucket.org

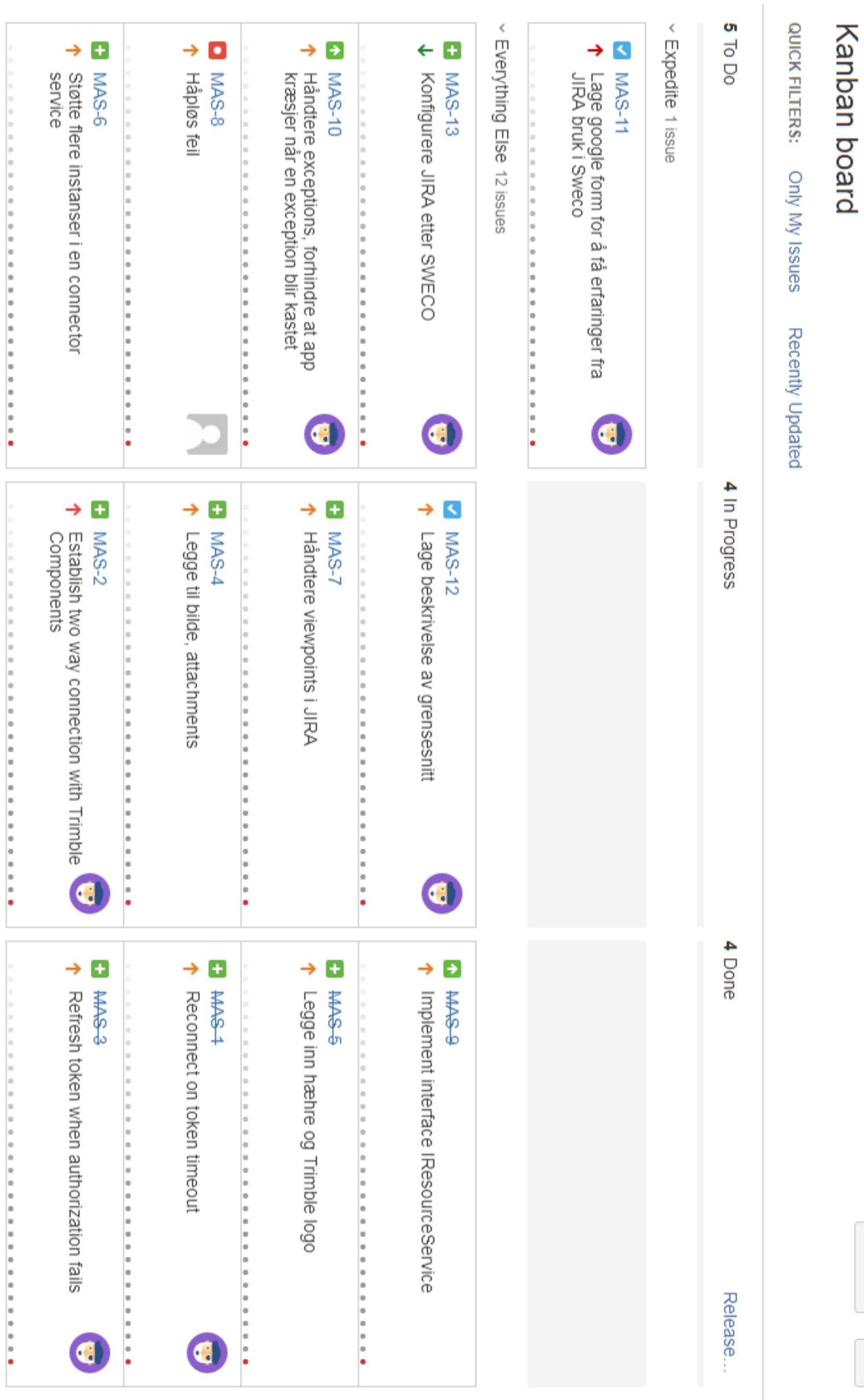


Figure 4.1: A typical Kanban board with a set of issues.

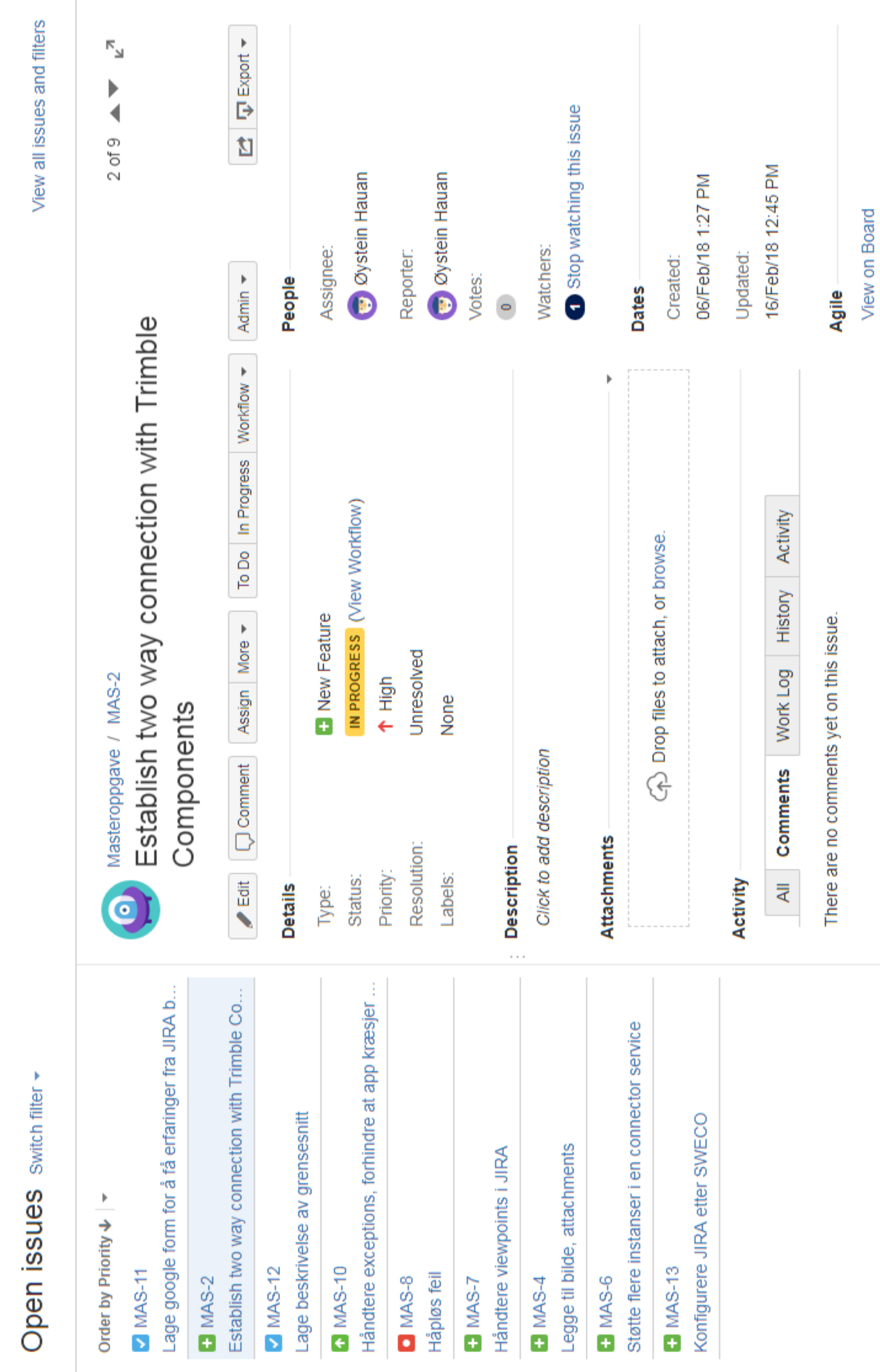


Figure 4.2: Standard issue view screen

responsible for the task, and it ensures that the right people are notified at the time of issue creation and when important updates are registered.

Figure 4.2 displays an example of a typical issue view in Jira for a software project.

However, for this project, Jira is to be evaluated as a tool for non-software related projects, and in particular for large, complicated infrastructure projects in the Norwegian industry. With this industry looking to adapt project management methods from agile software development, it is interesting to see what impact the implementation of software such as Jira can have on the industry. Jira was selected as the primary software for this project, because of a feature and technology analysis conducted in the project thesis prior to this thesis. An overview of researched technologies can be found in Appendix A.(*Although Trello has a higher overall score than Jira, Jira was a better fit based on user experience with both. Originally, this thesis aimed at connecting Trello, in addition to Jira, to the IssueConnector, but the time constraint did not allow for this to be done withing the designated time frame. However, seeing as Trello has a powerful API and supports WebHooks, it would be uncomplicated to connect it to the system.*)

Although Jira is designed specifically for software development projects, it is highly configurable in almost every single aspect of it's design, which indicates that with modifications, it could be a great option for other industries to adopt as well. Moreover, various Norwegian companies have already tested Jira in a few infrastructure projects. It should be noted that there is very limited data available on the success of these projects, as they are recently executed. A select few of the involved in these test projects have agreed to offer some thoughts on their impression of Jira as a project management tool in this industry. The answers to this questionnaire are available in Appendix H, and are discussed later in this chapter.

4.2 Requirements for Jira

If the infrastructure industry is going to adopt a new software like Jira, it is important to define some requirements to the software's functionality. The design phase in infrastructure projects is complex, with many different fields involved, meaning there is a substantial amount of features that must be supported.

The following user stories, requirements and user roles were defined as part of the project thesis preceding this thesis. They were defined at meetings with experts from relevant fields, primarily participants of the SPP project. This work has been the basis for this part of the thesis, concerning project management and issue tracking in the Norwegian infrastructure industry.

4.2.1 Roles, Stakeholders and End Users

In a session at Sandvika on November 2nd 2017, part of the agenda was to define user stories and/or use cases for the project management system. Some of the roles and user stories have been revised at a later time.

The first part of the meeting was spent discussing who should have access to and use the system. It was soon established that the system must support role based access control, meaning different users have different roles and access rights. Important users of the system were then listed, and they were as following (in no particular order):

- System administrator
- Design engineers
- Domain manager
- Project manager
- Design manager
- BIM-coordinator
- Facilitator
- Project owner, internal
- Project owner, external
- Official approving authority
- Site manager
- QA engineers
- HR engineers
- Environmental engineers
- **External stakeholders**:
  - Infrastructure users
  - Asset management engineers
  - Road user interest groups (e.g. NAF, Syklistenes landsforbund)
  - Property owners
  - Authorities at regional, state and municipal level.

4.2.2 User Stories

Having defined the most important roles in the system, it was natural to define some *user stories*; what some central actors might wish to do in the system. Please note that for a final product, more user stories would have to be defined and satisfied. However, to keep the scope of this project at a manageable size, only the most important stories are defined for now.

1. As a **planner/project designer** I wish to ask for advice or input from peer planners/project designers. I also wish to ask questions to domain experts in different disciplines. I wish to uncover faults and errors in the BIM-model to document and mark these.

2. As a **facilitator** I wish to view all issues and inputs, and be able to pass them along to the right people or assign the tasks. I also wish to estimate and prioritize all tasks in a session.

3. As a **project owner** I wish to view all events that have occurred, and browse through them. I also wish to be able to browse to a specific project phase. I want to be able to uncover challenges with solutions made in earlier phases, and I want to classify different decisions.

4. As a project **owner or designer** I wish to see all influential decisions made in earlier phases. I want to access a list of all change decisions.

5. As the **designer** I wish to be able to log the reasoning behind each decision. I want access to view the backlog for multidisciplinary tasks. I want access to view the preconditions and prerequisites for the project. I want to be able to communicate with my peers and receive notifications in real time.

6. As a **planner, project designer, coordinator or domain manager** I need to see requests for change and the dependencies between estimates in a structured view.

7. As **designer** I wish to be able to estimate the tasks that are planned for me to do.

8. As a **decision maker** I need all changes that have been made since the latest milestone/session in a clear and transparent manner so that I can manage them. I should also be able to request a decision.

9. As a **domain manager** I wish to see what my "people" have done, as well as what other domains or disciplines have done. I need a structured overview of all changes that have either been made or is going to be made, to define and view relations between them. I also want to be able to see expected tasks for an upcoming session.

10. As a **lead planner or facilitator** I need to be able to view available resources for a session.

4.2.3 Functional Requirements

Partially based on the user stories, are the following requirements for Jira. These were defined by participating experts in the SPP project.

1. Role-based access control

   - Different functions of the system should be accessible by users having a specific role in the project, and invisible/inaccessible to others.
   - Customized UI based on the role of the user.

2. Clear listing of relevant issues based on domain and user defined criteria.
   - See changes since last milestone or session. The user should be presented with issues and changes relevant to their role when they log in. This should appear as a dashboard, where the issues can be sorted based on different attributes of the issues.
   - It is important that the relevant issues can be sorted by status (default).

3. Intuitive UI
   - The UI should be simple to learn and use. It should be easy to use for both experienced computer users as well as those less experienced.

4. Configure issue type (decision, requirement, pre-requisite etc.)
   - It should be possible to specify what sort of issue has been created. An issue can for example be a bug, a decision, a requirement etc.
   - Conditional attributes based on issue type. Depending on what sort of issue is being created, custom attributes related to this issue, both optional and required, should be available.

5. Flexible templates for issues
   - It should be possible to define different templates for different issues. Designing these templates should be straighforward.

6. Categorisation and attachments
   - For a given issue, it should be possible to categorise it (bug, decision, pre-requiste etc.). It should also be possible to attach files (images, IFC-files, documents etc.).

7. Define set of attributes for each issue
   - It should be possible to define attributes for each issue, and classify these.

8. Configure notifications for mail [LOW]
   - It should be possible to receive notifications on e-mail, and customise which notifications one would like to receive.

9. Views for boards and estimates
   - One should be able to access a view that display s the sum of relevant estimations for reporting.

- It should also be possible to display/generate graphs of the estimates [LOW]

10. Project specific phase
    - It should be possible to navigate between the phases of the projects, both past, present and future phases.
11. Define dependencies between issues [HIGH]
    - It must be possible to define and display the dependencies between different issues, including one-to-one (1-1), one-to-many (1-n) and many-to-many (n-n).
    - It should be possible to generate and display a graph of said dependencies [LOW]
12. Log reasoning behind decisions
    - It should be possible to store the reasoning behind different decisions in a simple manner, and it should be clearly displayed when viewing the decision/issue.
13. Reporting
    - An important requirement of the software is to allow for reporting. It should be possible to easily generate reports based on custom requirements.
14. Log original estimate and remaining estimate
    - It should be possible to log estimates during the project phases, both original and remaining.
15. Ability to delete database records
16. Real time notifications (must be configurable).
    - You should receive in-app notifications in real time, based on your notification settings. You should be notified for example when you are assigned a task, when someone requests your opinion etc.
17. Ability to plan and log sessions/meetings
    - Tag the relevant persons who've attended the meeting
    - Tag/list decision makers
    - Generate graphs and histograms of decisions and statuses of issues
18. Assign tasks
    - It should be possible to assign a person to a task, either yourself or someone else in the project.

19. Web-based solution
20. Loose connection with the BIM-models through the BCF-format.
21. Export records
    - The application should be able to export the records in different formats, like PDF, CSV, JSON, XML etc.

4.2.4 Non-Functional Requirements

1. Price
    - The price is an important factor, as it should not be too expensive, because of the limited willingness to pay for expensive software within the infrastructure domain.
2. Setup and licensing complexity
    - It should be simple to add and remove users, and define their roles in the system.
3. On premise hosting
    - The application must have the ability to be hosted on premises, behind a firewall, to ensure security and restrict data leaks.
4. Security
    - In accordance with the above, the system needs to be watertight and secure.
5. Stability/reliability
    - The vendor should be reliable, and not be in danger of going out of service, due to bankruptcy or similar.
6. Performance
    - The system should be able to effortlessly handle up to at least 100 simultaneous users.

It should be noted that the non-functional requirements are limited to this project. In other words, if this project eventually is to turn into an actual system in production (as it most likely will), these requirements will have to be revised and strengthened to handle this. However, that has been determined to be out of scope for this project.

With the only exception being FR20 (FR3 is of course debatable), Jira supports every requirement specified. Not every feature is available out-of-the-box, but Jira offers an impressive add-on market, with over 1500 extensions available, both free and paid. If Jira is adopted widely in the industry, developing a plugin for importing and exporting BCF-files should not pose a major challenge.

4.3 Configuring Jira

There are mainly two ways of hosting Jira:

- Jira Cloud: This alternative is cloud hosted, meaning that Atlassian has the installation of the software, and the user subscribes to it. This eliminates the need for a dedicated server, but limits the amount of customization possibilities. There is no need for a dedicated database, and the pricing is subscription based.
- Jira Software (Server): This alternative is self hosted, meaning that there must be a dedicated server that runs the Jira software, and there must also be a dedicated database to the software (PostGreSQL, MySQL, Microsoft SQL Server etc.). This allows for a great amount of customization and configuration, and gives the administrator full control of the database. The pricing is a one-time payment for the software, depending on the amount of users, and an optional yearly price of maintenance and updates.

For this project, it was optimal to go for a self hosted version of Jira to achieve complete control of the software. This was necessary to make the programming and customization as simple as possible. Atlassian provided the project with a community license, which includes unlimited users and a perpetual license free of charge. This licence may not be used after the project is over.

4.3.1 Initial Configurations and Schemes

As this project primarily has been done at Trimble's premises in Sandvika, it was natural that the software was configured in their environment. Trimble already has a Jira instance, which they use for software development projects. The plan was initially to use this instance, or it's staging/test environment, to develop the integration application. However, it was soon apparent that to configure a Jira instance with the requirements of the SPP project, one would need full administrative access to the software. Since Trimble is a large, international company, obtaining the right licenses and access rights can be a slow and difficult process, so it was decided to install a separate version. Atlassian offers trial licenses for up to 90 days, and students can apply for community licenses which are completely free of charge.

With the help of the IT-department at Trimble, a server was set up with the sole purpose of hosting software, both external and self developed, relevant to this project. This is an ideal solution, as configuring services on a server often requires a lot of system tweaking and full system reboots. If the server is hosting important software, these could become temporarily unavailable, or worst case completely unavailable. Jira Software was installed on this server with a community license, with a MySQL database.

The Jira instance was installed as a service on Windows Server 2016 Standard Edition. This means that whenever the server reboots, the Jira service will start automatically. It was given a domain name and made accessible for all incoming traffic [6]. Since the instance is licensed with a community license, it has no limit for the amount of users.

The following configuration of Jira has not been properly tested, because the interest regarding the IssueConnector became significant early in the development phase, and the focus shifted more towards incorporating the software with the already configured Jira instances of the interested parties. This section of the thesis only presents a suggestion, or a starting point, for how Jira can be configured, based on Appendix B.

Schemes

Jira allows the administrator to create different schemes, that can be used across projects, or be project specific. These schemes can define what the user sees for each issue type (screens), who can see what (permission schemes), what statuses should be available (workflow schemes) etc. A large part of configuring Jira will be to define these schemes. Achieving an optimal configuration will take time, as people start to use Jira and get familiar with it, and provide their feedback. The most important schemes are:

- **Issue Type schemes.** These schemes define what issue type a project should implement. This means that all project can have different sets of issue types, or they can share the same schemes.
- **Workflow schemes.** This scheme defines the workflow of an issue type. A workflow in Jira defines the statuses an issue can have during it's lifetime, and also apply rules to the transitions between these. A standard workflow in Jira is *Todo, In Progress, In Review, Done/Closed.*
- **Permission schemes**. These schemes define what persons are allowed to view, edit and create issues in projects, based on their role in the project. This means that only the right people are allowed to create issues of a specific type, and issues that are confidential from some groups can be hidden from them.
- **Field configuration schemes.** These define what fields belong to each issue type in each project, allowing for full customization between projects.

A central feature to the schemes is that they can be reused for any new project. Defining and designing these schemes is therefore not something that has to be done each time a project is created. It does however require some initial configuration.

Issue Types and Custom Fields

Jira is as mentioned primarily developed for software development projects. These types of projects have a highly standardized set of issues, like *Bug*, *Feature*, *Enhancement/Improvement* to name a few. All of these standard types are available

[6]Link to the Jira instance that was used for this project: http://ic.novapoint.com:8080

out of the box for Jira, as seen in figure 4.3. But this terminology does not translate well to infrastructure projects. Jira allows you to create custom issue types to fit with the desired project. In Appendix B, a project log is proposed that is designed to contain all relevant data in a project. The issue types for this Jira instance are derived from this document, namely the ***Description*** column, and translated to English. The six issue types that were defined are:

- Final Decision (Vedtak)
- Action (Aksjon)
- Pre-requisite (Forutsetning)
- Temporary Decision (Beslutning)
- Assumption (Antakelse)
- Information (Informasjon)

The idea is that each of these issue types can have a custom specification of fields, both mandatory and optional, that separates these types from one another. A standard issue creation screen can be seen in figure 4.4. Most of the fields shown in that figure are standard. As with custom issue types, Jira allows for creating an indefinite amount of custom fields of various types (dropdown list, user selector, text input, radio buttons etc.). To begin with, the other columns from Appendix B were used to define these custom fields. An example of a custom issue creation screen can be seen in figure 4.5.

4.3.2 Configuring Jira for Betonmast Hæhre

Betonmast Hæhre are planning to start using Jira for their infrastructure projects. They are not familiar with the software, and it was decided that they should use a trial version of Jira for 90 days, and configure it to fit their needs as closely as possible. This trial version was installed and hosted at Trimble Sandvika, and the configuration was drafted in a workshop with Trimble, Cowi and Hæhre. The outcome of this session can be found in appendix D.

Inspiration

The main inspiration for the configuration of Jira, came from the test projects done by Sweco and Rambøll. They had a set of ideas that they had tried and tested succesfully, which they offered to share with the SPP-project. The reasoning behind this openness, is that the desired outcome for all involved parties is a new standard for process and task management across the entire infrastructure industry.

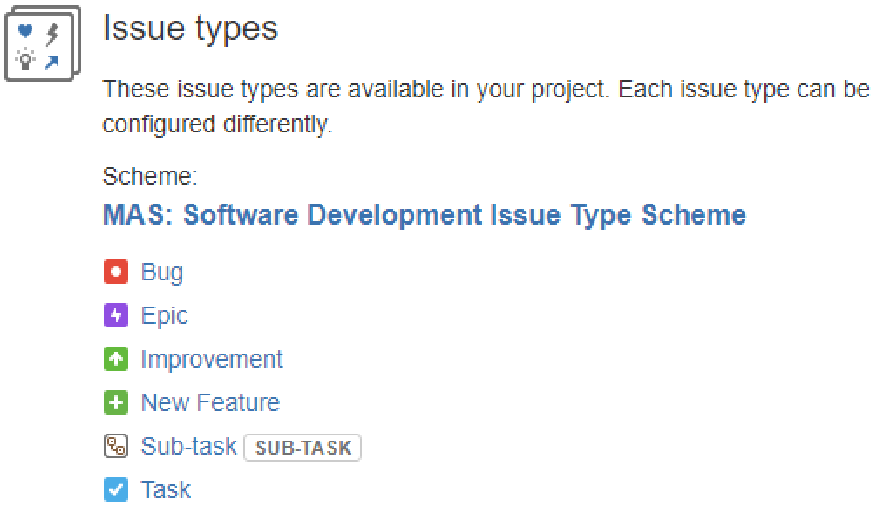


Figure 4.3: A typical set of standard Issue Types in Jira.

4.3.3 Issue Types and Fields

Supporting Hæhre's existing workflow was an important factor for configuring Jira. They wish to keep all their documentation within Jira. This instance of Jira was to be connected to one of their EasyAccess projects via the IssueConnector, and only some of the issue types should be synchronized. The issue types defined were:

- Meeting reports (Møtereferater)
- Conflicts and Comments to Model (Konflikter/Modell kommentarer)
- Internal decisions (Beslutninger, interne)
- External decisions (Beslutninger, eksterne)
- HMS/YM
- KS

Conflicts and Comments to Model are the main issue type to be connected with EasyAccess. These will automatically synchronize with the IssueConnector, as demonstrated in the demo video[7].

In addition to these, the standard issue type *Epic* is to be used as a type for transport corridors, or stretches of infrastructure, that the above issue types are a part of. The Epic type in Jira has a few special traits. It is not a single task, but

[7] https://drive.google.com/open?id=115Uz3gqcqIT8QpbYNk4UDPfjbMCKVkw0

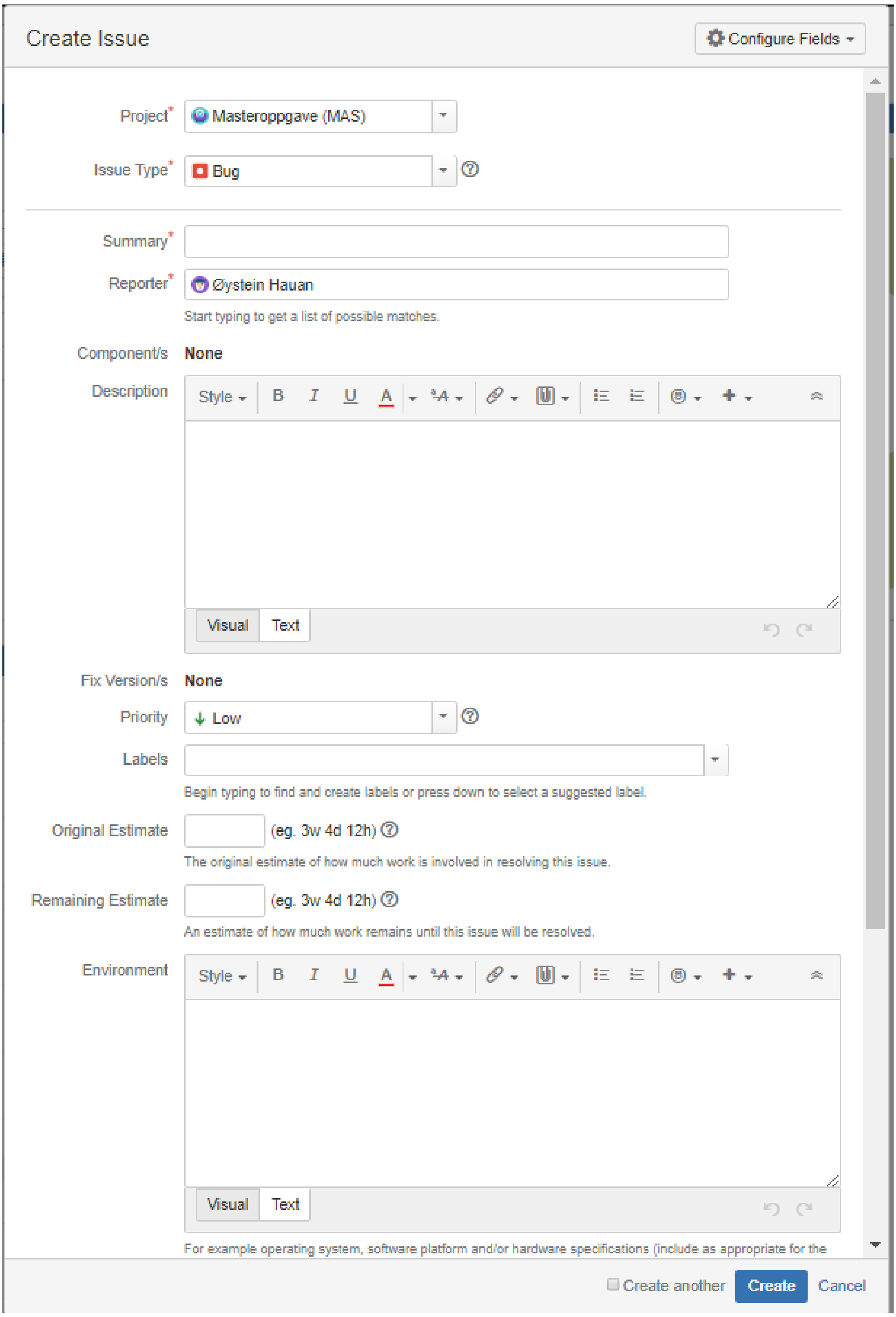


Figure 4.4: A standard Issue Creation screen in Jira.

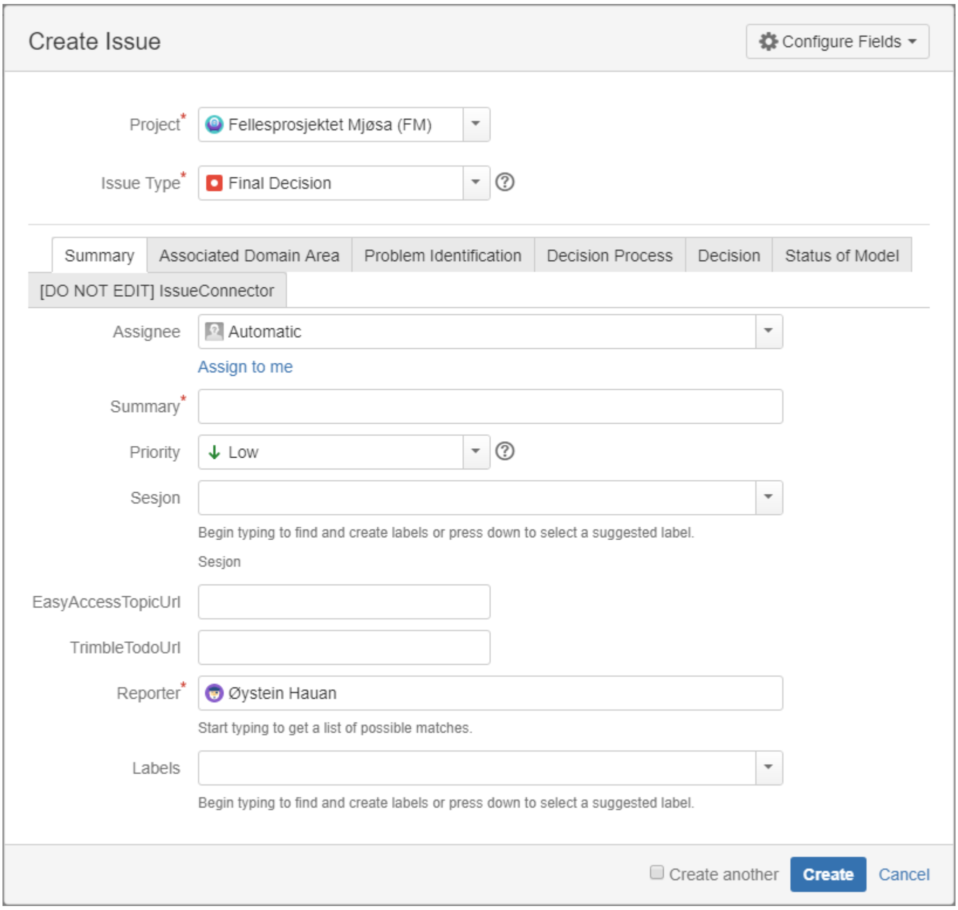


Figure 4.5: A custom Issue Creation screen in Jira. Notice that the the different tabs correspond to the categories in Appendix B

rather a large body of work that can be broken down into smaller tasks. This way, it is easier to estimate, track and manage the progress of a specific stretch of the contracts.

Jira has a special field for issues called *Component.* Atlassian defines components as : "[...] generic containers for issues. Components can have Component Leads: people who are automatically assigned issues with that component. Components add some structure to projects, breaking it up into features, teams, modules, subprojects, and more. Using components, you can generate reports, collect statistics, display it on dashboards, etc.[...] Nothing prevents users from adding issue to more than one component."[26] Components in this configuration are going to be used for the different disciplines an issue relates to. Examples are *Road, Water and Sewarage, HMS* to name a few.

Another special field in Jira is *Fix Version.* This field describes in which *release* of a software an issue is planned to be fixed. For this project, *Fix Version* is used as milestones; dates where tasks or work will be finished. A screenshot of a customized issue of type *Conflict* is displayed in figure 4.6. (*Please note that some of the information has been blurred out and edited to hide confidential information.*)

Hæhre are using *sprints* to plan which weeks to work on tasks. In figure 4.6, the related sprint can be seen in the bottom left corner, along with the end date of the sprint.

4.3.4 Workflows

Each issue type in Jira can have an individual workflow. The initial workflows configured for Hæhre can be found in Appendix D.

4.4 Future of Jira in the Norwegian Infrastructure Industry

Sweco and Rambøll have already started testing using Jira for infrastructure projects, and some of the involved agreed to answer a few questions in a survey, available in Appendix H. Hæhre Betonmast have tested Jira for a few weeks, and some of involved in this project shared some feedback and thoughts on future potential.

4.4.1 Evaluation and Experiences from Hæhre Betonmast

Appendix F lists an e-mail thread where a few of the participating experts give their thoughts and evaluations of the potential of Jira and the IssueConnector. Based on these e-mails, it is widely agreed upon that Jira has a great potential in the industry. It enables a more detailed, streamlined and simple process of planning a project, meaning that all decisions and tasks, both completed, in progress and to be done, are stored in one place. Combining this with the real-time synchronizations through the IssueConnector, the future seems promising, with Rasmussen stating "[...] The large payoff will become apparent once you manage the entire project through Jira".

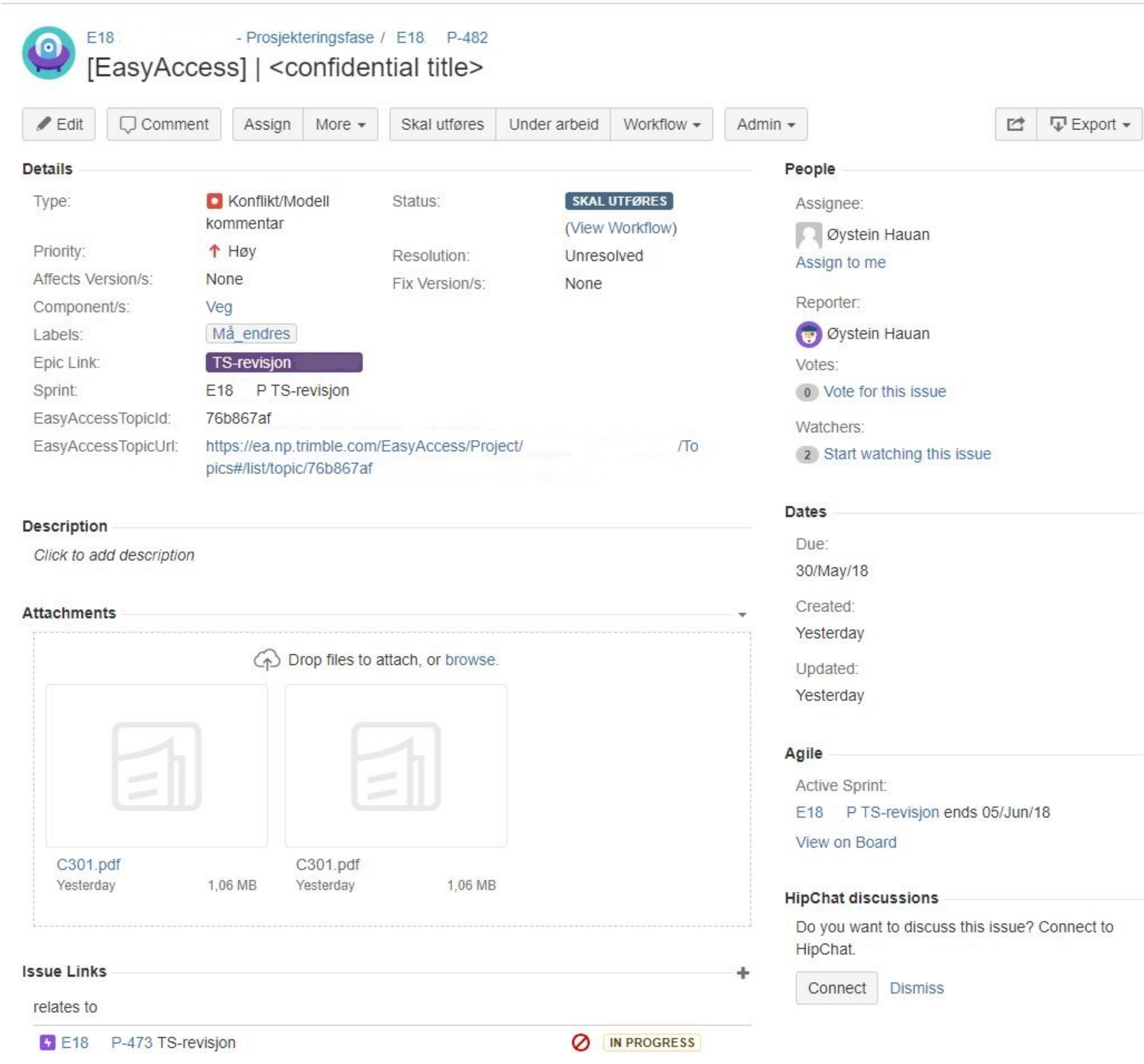


Figure 4.6: An issue in Jira configured for Hæhre

According to Jan Erik Hoel in appendix F, the design processes in the infrastructure industry are shifting from rigid waterfall processes, to a more iterative approach, resembling what the software development industry did many years ago. Jira is designed to support iterative project methods, meaning it could be a great choice for implementing such a tool in full-scale infrastructure projects.

There are however a few challenges that must be considered. One major difference between the infrastructure industry and the software industry are the people working in them. Software projects tend to be comprised of ”tech-savvy” experts, who are familiar with computers and software. The threshold for introducing new software to this group of people will in all likeliness be much lower. People from the infrastructure industry range from quite experienced computer users, to rather unexperienced. Implementing a new software tool for managing every aspect of the project is therefore much more likely to cause frustration and confusion among these users.

Kjartan Kristoffersen states that “The UI of Jira should be more intuitive. [...] Those who do not work directly with designing and who are not especially interested in technology need to see this as a simplification.” This is probably the major issue for introducing Jira or similar software today. There is an undeniable shift in the workplace, spanning more or less every industry, towards a more technologically driven work method. The infrastructure industry is severely lagging in this area, far behind the oil and gas industry. People in the industry who are keen to change the methodologies and processes, need to drive this disruptive change forward, but it will require patience and resource investment. People are generally reluctant to change. Consequently, the introduction of new work processes should be introduced in a convincing manner, enabling the users to experience the benefit of the software.

4.4.2 Results from survey

Appendix H displays the responses registered from a select few, who have had some experience using Jira in test projects in the industry. All of the respondents state that they are to some degree experienced software users, ranging from somewhat experienced to experienced. Only one of the respondents think the UI is overwhelming, and most believe Jira has potential in the industry.

The respondents come from different disciplines in the projects, which was an important factor to this questionnaire. Jira is to be used by everyone involved in the project from start to finish, which means that there will be users with all kinds of background regarding software and computer experience. This sparks the need for a simple UI, specifically designed for the person using it. This was also reported by one of the respondents:

“First impressions were a little overwhelming. A lot of text and little visual aids. After getting used to it I have appreciated to have all of the resources and tasks in one place, with the possibilities of commenting and sorting tasks in sprints. Yet I think it could benefit of a better user interface with less clutter/information and more visual aids.”

The first impression of Jira can indeed be overwhelming. There is potentially a *lot* of information, seemingly randomly placed around the dashboard. But an advantage with Jira is that it can be configured to look almost exactly as one would want for any logged in user. By defining project roles and permission schemes, administrators can decide which information should be available to whom, and what changes they can make. As previously stated, developing this optimal configuration is going to take time, adjusting settings based on feedback from the users.

Another respondent noted one of the main benefits of Jira:

"All project members are also given insights to the whole project, and to the challenges and workloads of co-workers". This is an important factor, that has been discussed earlier in this thesis. The transparency of a fragmented project can be severely improved, which can ultimately lead to better, and more readily available decisions. There is an agreement among most of the respondents that this, in addition to the easy way to define and manage tasks, is the main benefit of implementing such software.

Some of the drawbacks that are mentioned are, not surprisingly, that it will be challenging to implement. But this relates directly to what the SPP group is working towards - changing the process and design of these projects. It will be challenging, but it is necessary, and the government requires it.

The amount of tasks may become very large, and the UI may get cluttered. If the benefits of Jira are truly to be experienced, it requires rather strict following up of tasks. Tasks that are finished need to be marked as such, and they must be assigned to the correct people or group. This all relates to training and documentation of how procedures should be done. It will take some getting used to, but compared to endless, confusing e-mail threads, the workload will be considerably more manageable.

4.4.3 Challenges

In addition to the aforementioned challenges, there are a few that are Jira-specific.

1. **The concepts and terms in software development, do not directly translate to other industries.** Terms like *Epic, Fix Version, Story, Components* and so forth, are not used in the infrastructure industry. Even though Jira is highly configurable, some of these basic terms can not be changed or translated in the software. Sweco has made a guide to their users to explain what they use these concepts for, but it would be best to have a direct translation in the software.

2. **Jira requires an experienced administrator.** This is rarely a problem in the software engineering industry, but in this industry, it is not a given that a company employs someone with the necessary experience to administer this software. An alternative is to purchase Jira-Cloud instead, and rely on the developer to do the administration. This restricts functionality, however, and software such as the IssueConnector will not work properly.

3. **Cost.** This may not be a problem, but it is worth mentioning. Jira is not free. The fixed priced, self hosted version costs for 2000 users $26,400[8]. In

[8] https://www.atlassian.com/software/jira/pricing?tab=self-hosted

addition, to receive updates it costs half of this every year. It may not be much for some, but the price is definitely considerable enough to make it more difficult to "sell" in the industry. They do offer 90 days free trial.

4.4.4 Summary and Conclusions

The agile methodology in software development evolved in the 1990's as the industry was facing a crisis. Requirements and specifications changed faster than products could be released. An aerospace engineer from the 90's named Jon Kern, became increasingly frustrated with these long lead times and with the decisions made early in a software development project that couldn't be changed later [27]. This seems very familiar to the problems the infrastructure industry is facing today.

There is an undeniable need for progression in this industry. Jira delivers on almost every aspect of the requirements for such software. Almost every person asked who has tried it is positive regarding the potential of such software. An important bias to consider is that every person who has contributed to this project, is both interested and believes in changing the current methods. This means that the feedback from both Hæhre and the survey portraits only the views of experts who already believe in adopting this software.

However, few will disagree that the lead times for Norwegian infrastructure projects are too long, and Jira definitely offers opportunities in improving communication, planning, reporting, design, multidisciplinary work, task management, progress tracking and monitoring, to name a few. It is easy to imagine that these benefits will have an impact on lead times, and push the line further towards the objective of a 50% reduction in planning and design time.

Chapter 5

IssueConnector

5.1 Introduction

The larger part of this project has consisted of developing a prototype integration platform that can connect different services, and allow them to communicate seamlessly. This should all happen "behind-the-scenes", to ensure both simplicity for the user, and avoid incorrect usage. The solution should be as "open" as possible, in the sense that it should be simple to connect with any other relevant system, without it affecting the other systems in any way. For this project, the systems that have been connected are Jira and Trimble EasyAccess. EasyAccess (EA) is a web-based solution that allows users of NovaPoint (NP) to create issues (called *Topics* in EA), and connect them to so-called Viewpoints in the BIM-model. While EA supports registering issues, the ability to manage and follow-up these is highly limited. Therefore, Jira is to be used as the management tool, and they should communicate seamlessly in real-time. (A screenshot of NovaPoint with EasyAccess on the right is displayed in figure 5.1.)

Since powerful software such as Jira already offers much of the sought after functionality, it would make little sense to "reinvent the wheel" by developing another platform for the same purpose. The objective of the IssueConnector is rather to connect these existing systems, and with this expanding their area of application. EA Topics and Jira Issues are quite similar, taking input for Assignee, Author/Reporter, Type, etc. However, a Jira Issue can contain *a lot* more information than an EA Topic, meaning a common format must be established.

5.1.1 Development Specifications

- Operating System: Windows 10 64bit
- Programming language: C#, .NET Core 2.0.[28]
- IDE: Visual Studio 2017 Community Edition. *Note: the project has also*

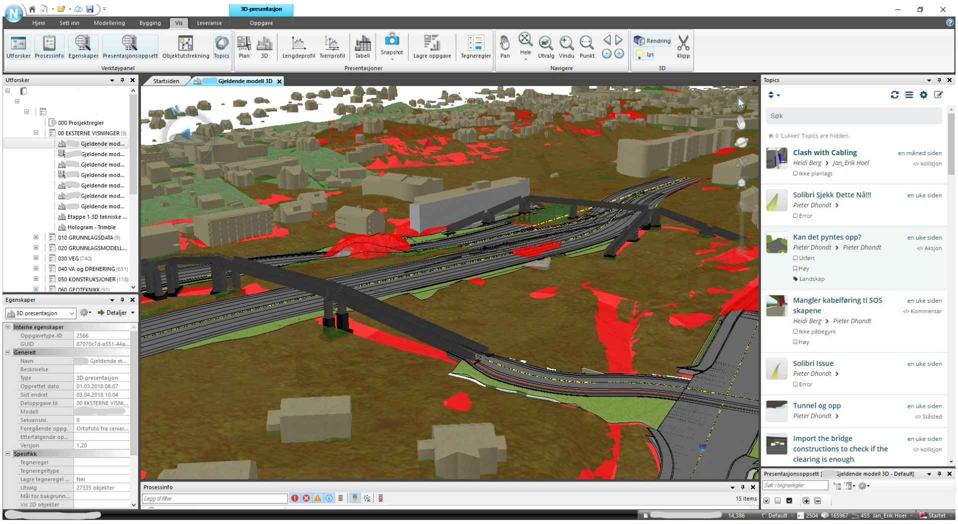


Figure 5.1: NovaPoint with EasyAccess.

partially been developed using JetBrains Rider[1]*, when coding on a Mac.*

- Version control: Git, Bitbucket (Private Trimble repository)
- Server Specifications:
  - OS: Windows Server 2016 Standard, 64bit
  - OS version: 1607
  - IIS version: 10.0.14393.0

5.1.2 Technologies and Frameworks

Following are a few short introductions of central technologies used during development. These are referenced throughout this chapter.

C#

C# is a general-purpose, object-oriented programming language. It was chosen because it is the most commonly used language at Trimble Solutions Sandvika, and thus simplifies both deployment and future development.

.NET Core 2.0

.NET Core 2.0 is a free, open source web framework. It is the "next-gen" ASP.NET, and is developed by Microsoft. It was chosen due to a few key features:

[1] https://www.jetbrains.com/rider/

- Cloud-optimized runtime, i.e. optimized for developing a web based, distributed system.
- Open-source.
- Cross-platform, i.e. it can be deployed on Windows, Mac and Linux.
- Easy to set up and start using
- Dependency injection[29].
- Simple deployment to Azure, which Trimble subscribes to.

Visual Studio provides several useful frameworks, or skeleton codes, of .NET Core, depending on the sort of product being developed (website, webservice, API etc.).

Visual Studio 2017 Community

Visual Studio 2017 Community Edition has been used as primary IDE for this project. This is Microsoft's free version of Visual Studio, and a natural choice for C# and .NET application development.

RabbitMQ

RabbitMQ[2] was selected as the message broker. It is an open-source, Erlang based broker, with more than 35.000 production deployments. The system is highly stable, easy to install and very powerful. It supports fanout messaging, which is needed for a Publish-Subscribe pattern.

MassTransit

MassTransit[3] is the chosen message bus for this project. It is open-source, developed for .NET and has extensive feature support when it comes to message handling. MassTransit is frequently updated, and makes it easy to implement Publish-Subscribe in a .NET environment.

Atlassian.NET SDK

Atlassian.NET SDK[30] is a free, open-source framework for easily establishing communication between a .NET application and the Jira REST API. It is developed by a QA-engineer at Atlassian, but is not official, unlike their Java SDK. It includes standard features to read, update and delete issues, and it is also possible to download the source code to extend features.

[2] https://www.rabbitmq.com/
[3] http://masstransit-project.com/

Trimble SDK's and NuGet packages

To establish communication with Trimble EasyAccess and Trimble Connect, various Trimble SDK's and NuGet packages have been used. These include: Trimble Connect SDK[4], and Trimble.Identity and Trimble.Diagnostics NuGet packages.

MySQL

For this project, the database used has been a community edition of MySQL. This database has hosted both the on-site Jira deployment database, and the shared database for the IssueConnector.

5.2 Architecture

Initially, the development of the IssueConnector was highly demo-oriented. In other words, the most important objective was to get a prototype up and running, allowing the features of the service to be demonstrated. It was easy to engage a large number of people in the development of this project. A prototype was launched quickly, illustrating how a least viable product version requires little time in development. It should also be noted that, at the time of project launch, the developer was unfamiliar with both C# and ASP.NET Core 2.0.

The initially rapid apporach lead to a somewhat chaotic code base. At this point, one of the main issues was that almost any error would cause a total crash, even for a minor communication error in one of the connected resources. Given the developer's limited experience with C# and Visual Studio prior to this project, many beginner's mistakes were also implemented in this early development version. Since project start, Trimble stressed that an important success criteria for this project is that the code should be intuitive and readable to easily enable further development by another developer. Thus, at some stage a shift in focus, from adding new features, to reqriting the whole applications, had to be made, in order to deliver a release-focused product. Continuing development on what was a typical case of "spaghetti code", would have been demanding for anyone other than the initial developer, and very difficult even for him. This lead to a significant period of time where no new features were added, in order to design an architecture that would be suitable for future deployment and development. As Barry Boehm stated:

"*If a project has not achieved a system architecture, including its rationale, the project should not proceed to full-scale system development. Specifying the architecture as a deliverable enables its use throughout the development and maintenance process*"

[31]

[4]https://github.com/Trimble-Connect/dotnet-sdk-docs

5.2.1 Designing the architecture

Software architecture has been defined in many different ways since the concept was first introduced in the late 1960s [32].

"*An architecture is a description of system structures, of which there are several (module decomposition, process, deployment, layered, etc.). Architecture is the first artifact that can be analyzed to determine how well its quality attributes are being achieved, and it also serves as the project blueprint. An architecture serves as the vehicle for communication, is the manifestation of the earliest design decisions, and is a re-usable abstraction that can be transferred to new systems. These are the things we mean when we use the word architecture.*"
[31]

"*The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.* "
[31]

The outcome of this project will, with high probability, lead to a product release, aiming to change the way infrastructure projects are planned and organized. This means that it is important that the IssueConnector not only displays the possibilities of such an application, but also defines a solid, maintainable basis for this product. Choosing the right architecture was therefore vital in this project. As the systems main objective is to be a connector between systems from different developers, there was a significant focus on interoperability for this architecture. The IssueConnector can be categorized as an integration system, meaning it's main purpose is connecting functions of sub-systems, and thereby adding value to the whole system. [33]

As mentioned, the initial approach to this project was highly demo oriented, focusing on functionality and features over quality and stability. After about six weeks of development, the IssueConnector started to gain attention from different people in the industry, who wished to test the system on their own projects. This defined a turning point for the development, as this would require the IssueConnector to be considerably more stable performance-wise than it was at the time. It became apparent that the software would have to be refactored significantly in order to achieve this. At this point, it was important to sit down and define exactly what the software should do.

System Requirements

In order to outline a system architecture, it was important to define a set of requirements for the software, and to prioritize these. This was done through several meetings, and the requirements evolved as the demo application took shape.

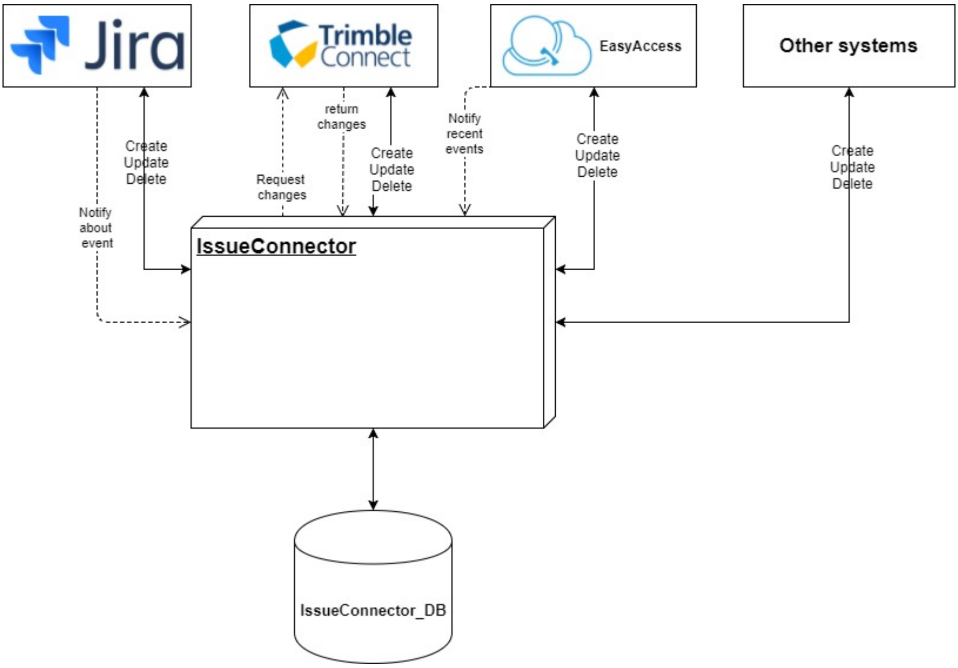


Figure 5.2: A simple overview of the IssueConnector.

Functional Requirements

1. An administrator should be able to connect projects from different systems through a user interface.
   (a) The interface should be a web page with login.
   (b) It should contain tools for setting up the service, including user mapping, field mapping, project mapping, defining information for connecting the systems etc.
2. When an instance is created at a given system, the service should create a similar, mapped instance in the connected systems.
   (a) The service should create the instance in the correct project
   (b) The created instance should display where the instance was originally created
   (c) The instance should contain the identifiers and direct URL's to the instance in the other systems. (If possible)
3. When an instance is updated in any of the connected systems, the service should register it, and perform the update in all the connected systems.

4. A project administrator should be able to customize the mapping of the fields in each system for each project.
   (a) The service should contain a default mapping for the different object models for standard fields, like status, priority, title etc.
   (b) The administrator should be able to access an interface where each field in each system, can be mapped to another field in a different system.
   (c) This mapping should be customizable for *every* project in each system.
   (d) A project mapping should be able to ”save” in order to be reused in similar projects.
5. The service should be able to handle attachments and upload these to all the connected systems.
6. The service should be able to handle viewpoints, and display these in an appropriate manner in all connected systems.
7. The service should handle comments, displaying them in all systems where comments are enabled. The comment should display where it was originally posted, and by whom.
8. The service should map usernames across the connected systems, so that all functionality with assignees, authors etc. is retained throughout all the systems.

Non-functional Requirements

1. The service should be a webservice that runs continuously, and connects the required systems.
2. It should have high availability.
3. It should be loosely coupled, to ensure that the downtime of one service does not affect any of the others.
4. It should be simple to connect new systems to the web service.
5. The web service should update the connected systems in real-time.
6. It should translate and map different objects/data types, so that the different systems will work together seamlessly.
7. It must ensure high reliability, to avoid data loss.

8. The service should be developed with maintainability and readability in mind, to ensure the possibility for further development (by other developers).

9. The service must be able to handle a large amount of requests.

10. The service should be scalable in order to enable future expansion.

11. The service should be as stateless as possible, to ensure that each instance/message is handled independently, so that errors don't propagate.

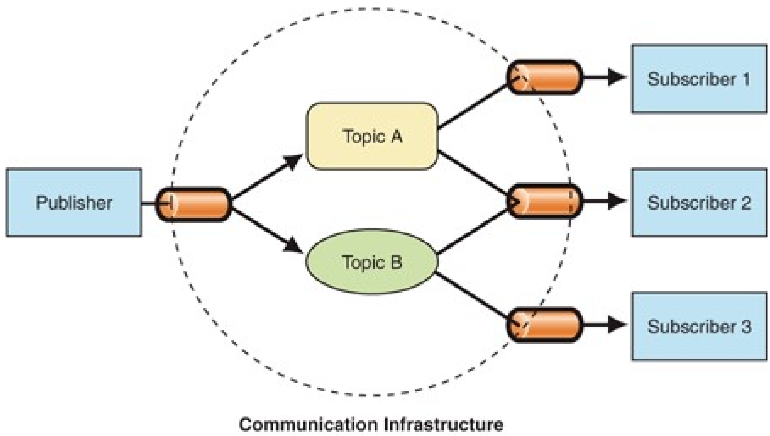


Figure 5.3: Publish/Subscribe pattern

The first two requirements state that the service should have high up-time and reliability. As with most web services, a user could need the system at any time, requiring it to always be online. Reliability implies that the service operates in the manner that is intended, and does not corrupt or misplace data during the process. This requires message filters and event handlers, ensuring that the right operation is performed at the right place in the right time. The data processed by the system could be sensitive, and must be treated as such.

In software development, a common requirement is to ensure loose coupling between the parts of the system. This has several advantages:

- **Separation of concerns.** This refers to a design principle, where the overall goal is to split the system into several parts, where each part has a specific objective. This is important for the system as a whole, but also the design for each individual service. Traditionally, many systems have been designed in a way that could be represented by one large box, with no clear structure or message flow, which is a sub-optimal solution.

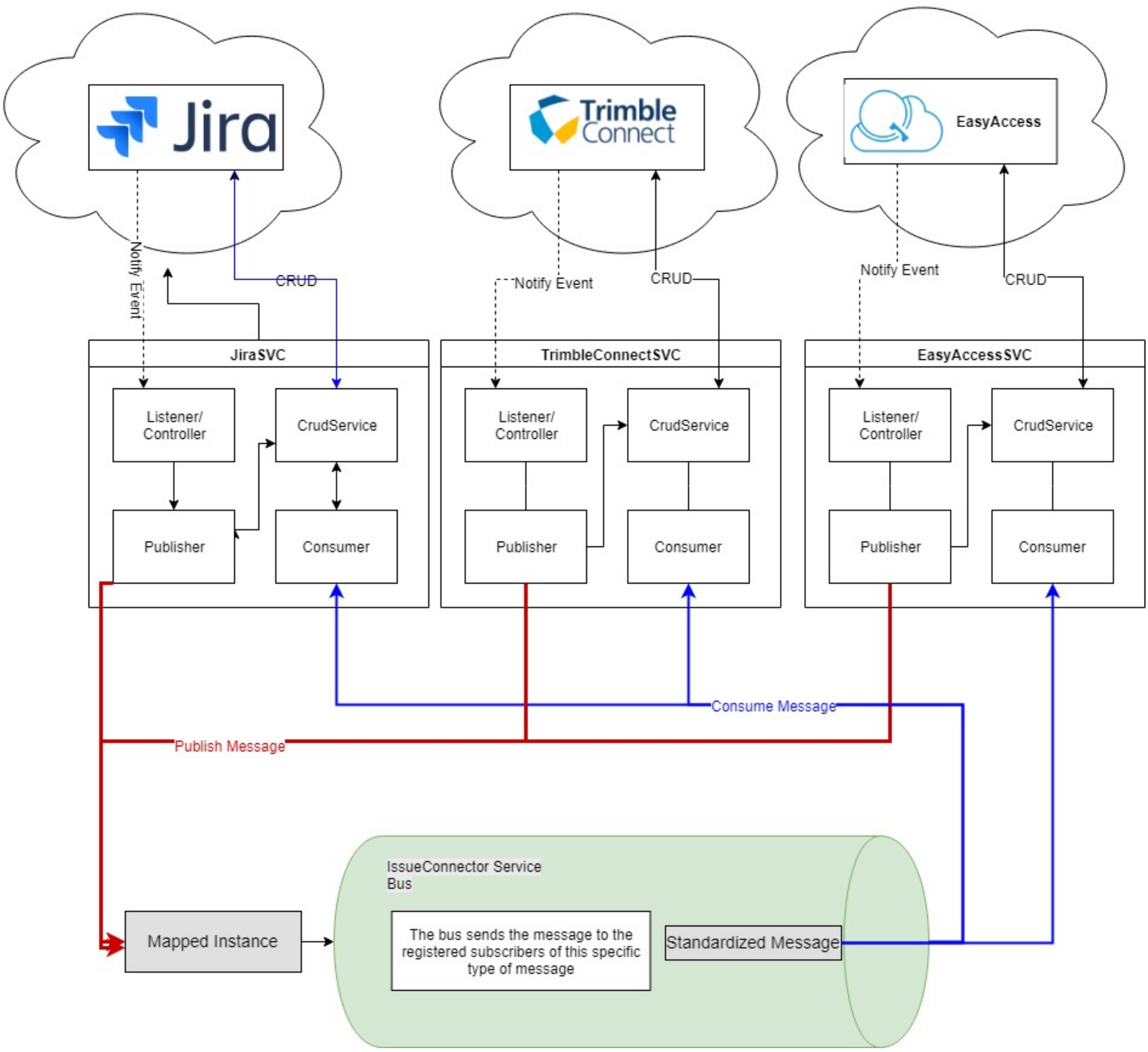


Figure 5.4: Full architectural design of IssueConnector

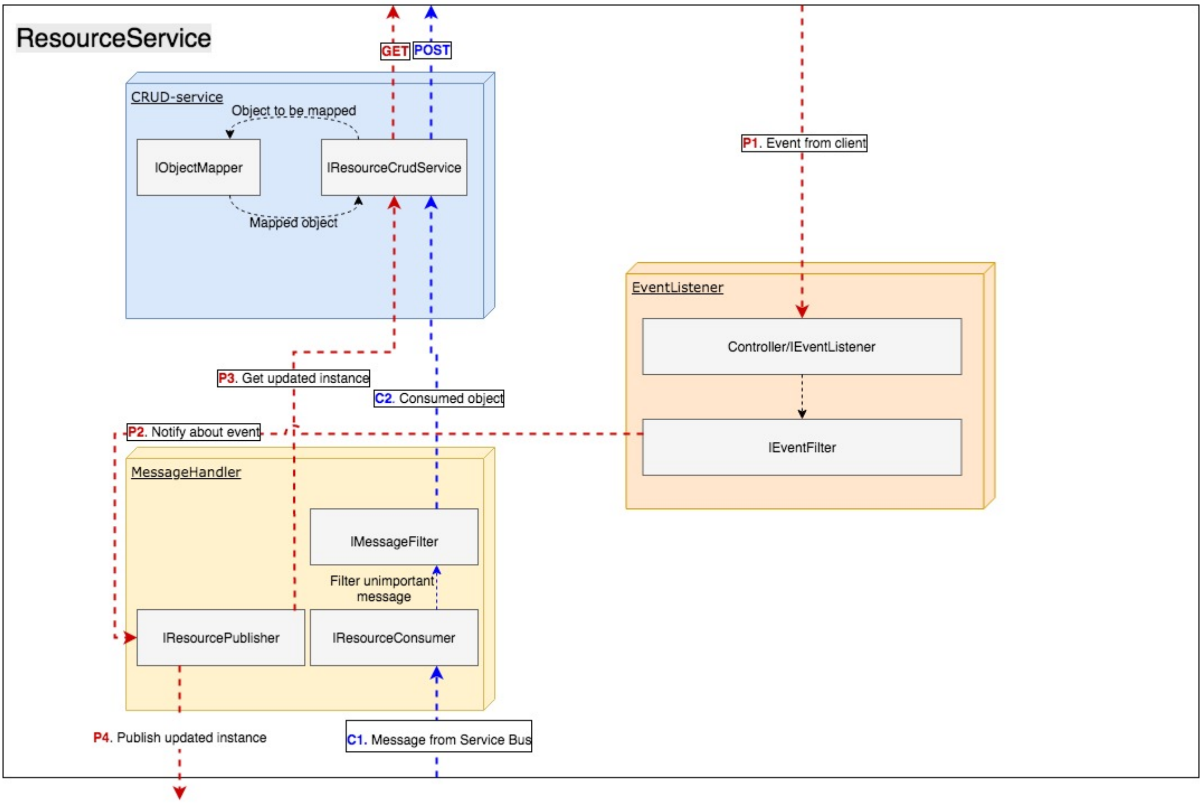


Figure 5.5: Service architecture and message flow.

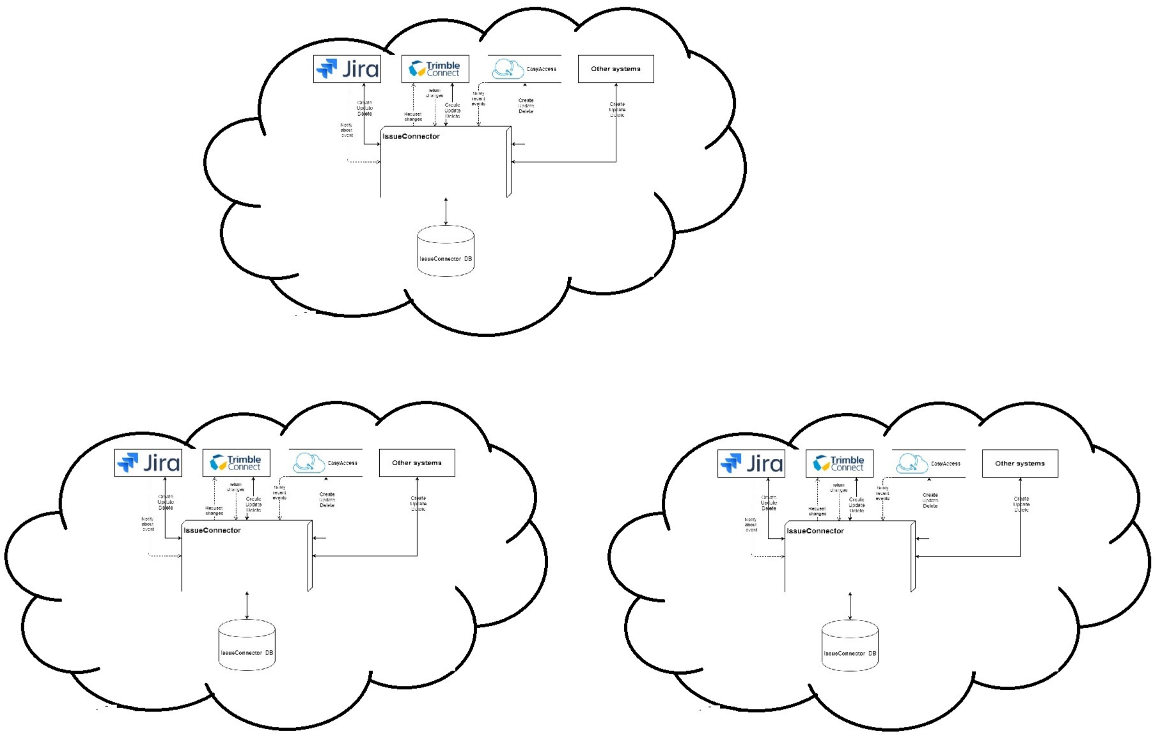


Figure 5.6: Cluster of IssueConnector systems.

- **Cleaner message flow.** Spaghetti code is a term used for unstructured and highly unmaintanable code. There is typically no clear, strict message flow in the system, and every part of the system is potentially dependent on any other. This increases the chances of failure, as the code does not adhere to certain design standards, or good practices, made to increase the performance and reliability of software. It does not so much concern what a system does, but rather how well it does it.

- **Reduce bottlenecks and single point(s) of failure.** In a system that requires high availability, it is important to eliminate or minimize the occurrence of bottlenecks and so called singe points of failure (SPOF). A SPOF is a part of an application that breaks operation of the entire system when failure occurs at this point.

As the list of requirements grew longer, the need for a proper architecture increased. The non-functional requirements are the most important requirements to consider when designing an appropriate architecture. One of the most important requirements to this system is the non-functional requirement 3 (NFR3), saying that the system should be loosely coupled, in order to drastically reduce the chances of a single point of failure (SPOF). This is also directly connected to NFR2 [34]. It is vital for this service to be available at all times, to ensure that no data is lost in transactions. A strategy for achieving this, is designing the service in a way that eliminates bottlenecks that can break operations of the entire system, or at least strongly reduce the chances of that happening. Ideally, each service should have no knowledge of the other parts of the system, so if one goes down, the others are not affected.

Choosing an Architecture

As described in chapter 2, the problems of the IC were discussed with an experienced software architect. Focusing heavily on NFR2 and NFR3, he suggested implementing an architecture based on the Publish/Subscribe pattern, a well known design pattern used for similar systems. It is a viable selection for a following context: "*There are a number of independent producers and consumers of data that must interact. The precise number and nature of the data producers and consumers are not predetermined or fixed, nor is the data that they share.*" [31]. This describes more or less exactly the objective of the IssueConnector.

The Publish/Subscribe pattern, displayed in figure 5.3, is simple enough to understand. It consists of an interface with two roles; publishers and subscribers. The publisher does not know who is interested in it's announcements, and sends it's messages to a message broker through a message bus. An indefinite number of subscribers can tell the broker that it wants to subscribe to all events of a certain type. Different subscribers can subscribe to different events (or Topics in figure 5.3). For example, Subscriber 2 in 5.3 subscribes to both topics A and B, while Subscriber 1 only subscribes to topic A. The publishers and subscribers never interact directly, which is key to achieving the loosely coupled, distributed system the IssueConnector is designed to be. This way, it is much simpler to connect a new service to the IssueConnector, by simply defining what events it should listen to, and what events it should publish.

As previously mentioned, the initial development of the IC was demo-oriented, and when the decision of refactoring to a new architecture was made, it was important to develop a strong design that accommodates these requirements. Figure 5.4 describes the overall design of the architecture, and shows the basic message flow of the system. Each service implements both a publisher and a subscriber/consumer. The publisher posts any events it is programmed to distribute, and the consumer continuously listens for any messages of interest. The publisher and consumer are not connected to each other. The message flow through each service is described in figure 5.5. The blue arrow represents the message flow *from* the broker, and the red displays messages going *to* the broker.

The clouds represent the cloud based systems the IC is connected to via their respective API's (REST-based). There is a two way communication established between each system and service, by so-called CRUD (Create, Read, Update, Delete) operations. This allows the service to perform the required actions on the system when it receives notification of an event. The stapled line represents this notification from the connected system. There can be different methods for communicating that something of interest has happened in a given system. For example, Jira has the ability to send messages via so called WebHooks[35], allowing the IC to be notified whenever something happens. EasyAccess, however, does not have this functionality, but instead notifies the service via SignalR communication [36]. Systems have different ways (if any) of communicating events, and this must be implemented in the systems dedicated service in the IC.

The large, green tube at the bottom represents a message bus. A message bus "*is a combination of a common data model, a common command set, and a messaging infrastructure to allow different systems to communicate through a shared set of interfaces.*"[37] Defining standard object models, or contracts, for the message bus, allows the different connected services to communicate with each other, without being aware of each others existence. Achieving this is crucial, as the ability to add an indefinite amount of services to the IC is a requirement. If existing services must be adjusted for each new service, the amount of work needed to add a new service will become insurmountable, as displayed in figure 5.7.

5.2.2 Implementing Important Design Attributes

As previously mentioned, ensuring high reliability and availability (and also maintainability), shortened RAM, is of utmost importance in a system such as this, as it will have major impact on the Life Cycle Costs (LCC) of a developed system [38]. This means that a significant amount of resources should be dedicated to ensuring this. It may slow down initial development, but will in almost all cases provide a positive return on investment.

Availability and reliability are often closely connected. While availability describes the percentage of time a system is available for use, reliability describes how reliably it completes it's tasks. For example, the IssueConnector could have an availability of 99.999%, which may seem impressive, but if it is unreliable and mishandles operations, it won't really matter. Vice versa, if it is bulletproof and handles all operations correctly, it won't matter if the system frequently becomes unavailable.

Focusing on these attributes through the development process is vital. The system being developed will handle sensitive data and failure would be catastrophic. The Electronic Industry Association and U.S. Government have developed a standard that defines four processes during development [39]. The first two are important for this project, while the latter two are important for releasing a product.

1. Understanding user requirements and constraints
2. Designing the software for reliability
3. Ensuring reliability for production environments
4. Monitoring during operation and use

The first task is then to understand the requirements that were defined in the previous section, mainly the functional requirements. For this section, the focus will be on functional requirement 2 (FR2) and FR3, as these describe the essence of what the IssueConnector should do.

When an instance is created in Jira, the IssueConnector should instantly create a copy of this instance in EasyAccess (and any other connected systems). It must create this instance in the correct project. In a real world situation, projects from several different companies could be connected to the IssueConnector. If sensitive data suddenly appeared in the wrong system, it could lead to legal trouble and distrust from customers. There are several ways to ensure this:

1. **A separate service for each instance of a system.** In figure 5.4, it is displayed how each cloud system has a corresponding service. There will be several instances of Jira (and also other systems, but Jira will be used here as an example) connected to the IC; both Rambøll and Hæhre have separate installations of the software. This means there are two choices of how to set up the architecture. The first is having only one Jira service, which handles all messages for every installation of Jira connected. But this solution poses two main issues:
   - (a) A SPOF for all Jira instances is created. If one Jira instance is faulty, this could potentially affect all the others, which is unacceptable.
   - (b) We increase the chance of the service sending messages to the wrong Jira instance. If there was only one Jira service, it would have to contain access credentials for every Jira instance connected to it, and it would be connected to all these instances at the same time. It is not difficult to imagine a scenario where an identical project identifier exists in more than one Jira installation, and the message ends up in the wrong one. If such a scenario were to happen, the reliability would be non existent, and customers would not want to use the software.

A second alternative is to deploy a new service for each Jira installation. This may be more difficult deployment-wise, but far from impossible. If every installation has it's own dedicated service, ensuring reliability becomes simpler. Each service handles only messages that belongs to it, and contains no information on how to connect to any other Jira installation. If one service fails, the others will not be affected.

2. **Setting up a new cluster of services for each customer.** In figure 5.4, all the services are connected through one service bus. Since the publish/subscribe pattern is designed in a way that every subscriber receives all events of a certain type, every service connected to the IssueConnector will be notified whenever *any* service publishes a message. All the messages being sent through the message bus are standardized, as explained later, which means that when an update is published in e.g. EasyAccess, all the Jira services will be notified, even though the message doesn't belong to that customer. This can be solved by deploying a new cluster of services for each customer: Adding a new service for each connected system that customer wants, and deploying a dedicated message bus and broker that will handle only the messages belonging to that customer. Figure 5.6 depicts this solution, where each cloud represents one cluster of systems.

3. **Encrypting messages.** Deploying a new cluster for each customer could become difficult and expensive if the number of customers using the IC becomes very large. Another option could then be to incorporate end-to-end encryption, which means that only the subscribers with the right "key" are able to decrypt messages coming from specific publishers. This way, one can use fewer message buses, but it may be more difficult to implement.

In addition to knowing that the IC operates correctly, it should be ensured that it is available at all times. This means that if a unexpected error occurs handling a message, it should not affect operations in either the same service or the others. NFR11 states that the IssueConnector should be as stateless as possible, which is a key solution to ensuring availability in this case. A stateful program will remember preceding events, which can change the internal state of the program continuously. [40] A stateless program handles each request, or message in this case, independently, and after execution (or failure), there should be no traces of the message left in the program. In other words, the state of the program should be exactly the same before and after handling a message, which is how the IC should operate.

This is implemented in the IC by using a framework called MassTransit[5], which is a .NET message bus framework. Every time a message is consumed, the IssueConnector handles it based solely on the content of the message. The message flows through the system as described in figure 5.5, and if an error occurs it is logged, and the framework ensures that the service continues to consume messages, while also notifying the broker of the error. The message bus registers a queue for all

[5] http://masstransit-project.com/

the exception messages from each service, which an administrator can use to debug the errors. The same applies for publishing messages. If the IC fails to publish a message, the framework ensures that the broker is notified of the error, and the IC continues normal operation.

The broker used for this project was RabbitMQ[6], which is also open source, and easy to use with MassTransit. When a new subscriber subscribes to a message type, the broker generates a new queue exclusively for this subscriber. When the subscriber reads the message, the message is removed from the queue. If the subscriber is unavailable, the broker will add messages to the queue until the subscriber is available again, and the subscriber will receive them in the order they originally were posted. This way, the subscribers and publishers become completely independent of each other, which is a major advantage in this type of system.

For this project, only the first of the above listed solutions has been implemented, as development was limited by a short time frame.

5.2.3 Mapping and Standardizing Objects

The architecture design for the IssueConnector presumes that the connected systems have a common data model and command set, in order for them to communicate. A relevant example of this is the BCF-format, which is a defined standard for exporting, importing and transporting BIM-models and their accompanying information. Even though the BCF-format contains information about a task, including among others status, type, assignee, comments, viewpoints and coordinates, it is severely limited when it comes to more complex task management [1]. For this project, it was necessary to define a viable new standard to which all connected services must adhere to. For this prototype of the project, the goal has been to connect primarily Jira and Trimble EasyAccess, because an integration between these two systems has been the most sought-after from experts in the industry. In addition to these, Trimble Connect has also been part of the development, though less prioritized because of it's (current) limited usage.

Figure 5.2 is a simple drawing of the IssueConnector and its connections. The systems in the top of the picture are all different, each having it's own data models. For example, Jira uses *Issues*, described in chapter 4.3.1. It can hold an indefinite number of custom fields that are defined by a Jira-administrator. This means that there is no way to know *exactly* what this data object will contain, from the perspective of the IssueConnector.

Similarly, the other systems have their own data object models. Trimble EasyAccess uses *Topics*, which are similar to Jira-Issues, but contain fewer fields, and have no support for custom fields. Trimble Connect uses *ToDo's*, which again has it's very own data object. This inconsistency between the systems propose a significant challenge, as to how these systems will "talk" to each other, when they don't speak the same language. The messages need to be translated, or mapped, between these models, so that each system can produce useful objects in synchronization with each other.

[6] https://www.rabbitmq.com/

This is challenging, as the intent of the system is to make it possible for other systems to connect to the IssueConnector, without having to do any changes to the systems that have already been implemented. At first, the way these objects were mapped were one-to-one. In other words, when the IC received a notification about a new issue being created, the IC would map this instance to both the Trimble Connect ToDo, and EasyAccess Topic separately. This lead to unmaintainable code, with methods like this:

```
public async void CreateTrimbleToDoFromJira(string issueKey)
{
Issue JiraIssue = await
    JiraService.BaseClient.Issues.GetIssueAsync(issueKey);

    _logger.LogDebug("IssueKey from Jira: " + JiraIssue.Key);

    Todo newTodo =
        JiraToTrimbleConnectMapper.CreateToDOonJiraIssueCreation(
            JiraIssue,
            JiraService.BaseClient,
            issueKey
    );
    try
    {
        var JiraColumnName = _projectMapColumns.Value.Jira;
        var tcPid = (await _dbService.GetProjectMappingFrom(
                JiraColumnName,
                GetProjectKeyFromIssueKey(JiraIssue.Key.Value)))
                .TrimbleConnectProjectId;

        var trimbleConnectProject =
            GetTrimbleProjectClient(tcPid);

        Todo createdTodo = await trimbleConnectProject
        .Todos.CreateAsync(newTodo);

        JiraIssue.CustomFields.Add("TrimbleConnectLabel",
            createdTodo.Label);
        JiraIssue.CustomFields.Add("TrimbleConnectID",
            createdTodo.Identifier);
        await JiraService.BaseClient.Issues.
        UpdateIssueAsync(JiraIssue);
    }
    catch (ProjectNotPresentException e)
    {
        _logger.LogDebug(e.StackTrace);
    }
}
```

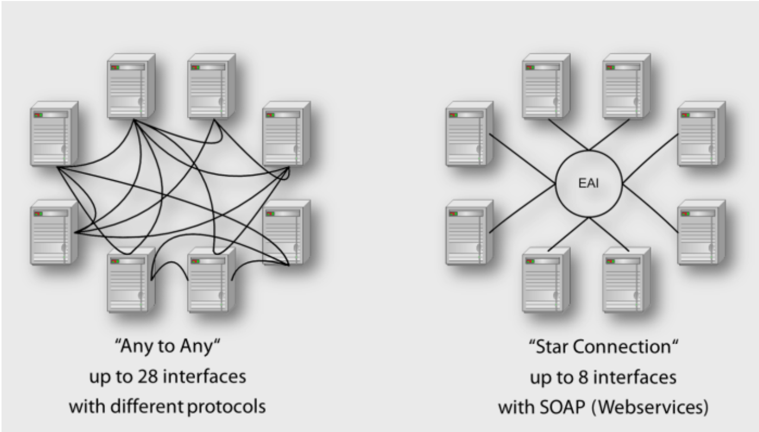


Figure 5.7: Diagram of point-to-point spaghetti structure [2]

The main problem with this approach, is that for every system to be used with the IssueConnector, a mapping function for all the other systems to the specific system would have to be made. For example, for Jira, TrimbleConnect and EasyAccess, the following methods would be needed:

```
Todo FromJiraIssueToTrimbleConnectTodo(Issue JiraIssue);

Topic FromJiraIssueToEasyAccessTopic(Issue JiraIssue);

Issue FromTrimbleConnectTodoToJiraIssue(Todo tcTodo);

Issue FromEasyAccessTopicToJiraIssue(Topic topic);

//etc....
```

This is not a viable solution, as the number of required mapping functions will become completely unmanageable as the number of connected systems increases. Figure 5.7 describes the exact same problem (except that the IC uses REST instead of SOAP), and a solution to it. The IssueConnector needs a common data format that all connected systems adhere to. Consequently, all objects related to specific systems need only be mapped to and from the common data format, which was named a *StandardInstance*. This is a common approach in Enterprise Application Integration[41].

As this project initially aimed at connecting Jira with EasyAccess and Trimble Connect, the StandardInstance was designed as a common ground between these object models. Several fields are more or less identical, like identifiers for the instance and corresponding project, status, priority, title, creation and modification date, author, assignee, and type. Type in this context refers to a classification of the instance. In Jira, the type would be (for a software project) *Bug, New Feature, Improvement* etc. The StandardInstance needs, at a minimum, to implement these fields. Since the Jira-Issue is the most verbose object model, it is natural to model the StandardInstance after this model, allowing most of the fields to be empty (or *null*), and allowing each translator to decide which fields are of interest and which are not. The interface defined as the "contract" between the systems is currently:

```
public interface IStandardInstance
{
        Dictionary<string, Identifiers> Identifiers {
            get; set; }
        string MessageOrigin { get; set; }
        string Action { get; set; }
        DateTime Created { get; set; }
        ClientUser Author { get; set; }
        ClientUser Assignee { get; set; }
        ClientUser ModifiedUser { get; set; }
        string Summary { get; set; }
        string Type { get; set; }
        string Priority { get; set; }
        string Status { get; set; }
        string [] Labels { get; set; }

        //The CustomFields map is a possible solution
            for
        //containing all the fields for an object.
        //This requires reducing the key and value to
            be of string type.
        Dictionary<string, string> CustomFields { get;
            set;}
  }
```

The field *Identifiers* is a dictionary that contains the identifiers of the connected instances in each system. When a Topic is created, the JiraService gets a message containing the StandardInstance that is a mapping of the created Topic, and creates this in Jira. The created issue then contains the unique ID of the Topic, and returns a new StandardInstance, where the *Identifiers* contain the ID's of both the Topic and Issue. For every system that creates an instance from this StandardInstance, the respective ID of this instance will be added to the *Identifiers* dictionary. This dictionary contains the instance ID, project ID, and also the URL to this specific instance in each system.

MessageOrigin is a field that contains a unique ID of the system and server that the message has originated from. *Action* refers to the action of the performed on the instance, namely CREATED, UPDATED or DELETED.

Author, Assignee and *ModifiedUser* are of type ClientUser. Since these are entirely different systems being connected, it cannot be assumed that the user information (username, email etc.) is equal across the systems. This sparks a necessity for a mapping of the usernames in each system, so that the correct identifiers are maintained throughout the systems. The ClientUser type is a temporary solution, which simply contains a mapping of the username in Jira, EasyAccess and TrimbleConnect. These are, for the moment, stored in a shared database.

A very important note to this technique is the responsibility of each service that connects the system. It is vital that the messages going out of a service are mapped correctly, so that the subscribers that receive this object can interpret them. No assumptions are made in each service about incoming objects. If different mappings are needed for different objects, each object should have it's own type. For this project, the types being interchanged are *IStandardInstance, IStandardComment, IStandardViewpoint, IStandardAttachment* and a map of identifiers called *IStandardIdentifierMap*.

For this project, the mapping is programmed in each service, with required methods defined in the interface:

```
IStandardInstanceMapper<STANDARD_OBJECT, SPECIAL_OBJECT>
```

This is a generic interface[42] where the service defines the object types going in and out of the mapping functions. For example, for the Jira-service, this would be:

```
IStandardInstanceMapper<IStandardInstance, Issue>

//the following methods have to be implemented:
IStandardInstance ToStandardObject(Issue issue);
Issue ToSpecialObject(IStandardInstance instance);
Issue ToUpdatedSpecialObject(IStandardInstance instance,
    Issue oldIssue);
```

Here, *Issue* is the object model of a Jira-Issue, used in the Atlassian.NET SDK[30], and *IStandardInstance* is the standardized object previously described.

5.2.4 Filtering and Authenticating Messages

Incoming Messages

An important part of systems that use messages to communicate, is applying filters that screen the messages and decide if they should be granted access to system. In a publish-subscribe pattern where all messages are received by every subscriber that listens to the corresponding message type, this becomes especially important. This is also very important for security reasons, to ensure that the system is resistant to infiltration attempts.

For example, every system connected to the IC at this point (Jira, EasyAccess, Trimble Connect) require authentication for the requests to gain access to the API. This is usually done using token based authentication[43]. A message includes a *Header* that contains information about the message; content type, origin, encryption etc. The header can also contain an authentication token, that is acquired through a process of logging in through a portal and receiving an access token. This way, a system can instantly evaluate if a message should be granted access.

At the time of this publication, the IssueConnector does not require any authentication, as it has only been used in testing environments. It does however implement a simple message filter that can be seen in figure 5.5. This filter reads the origin of the message, and evaluates if it is of interest for the respective system. For example, if a Jira-service receives a message with origin from a Jira system, it will dispose the message. Since each service is both a publisher *and* a subscriber, it will also check if the message came from itself, and in that case dispose it.

Outgoing messages

When an update occurs in a connected system, it should not necessarily be posted to the IssueConnector. Imagine a scenario where an issue has been created in EasyAccess. The respective service receives notification of this creation and posts a message to all subscribers. The connected Jira-service receives this message, and creates a connected issue in the Jira-instance. But then the Jira-service will be notified of this creation, and post a message to all subscribers that an issue has been created. The EasyAccess service will receive this message, and create another, duplicate issue. Unless some filter is implemented for outgoing messages, this loop will continue indefinitely, creating an insurmountable amount of duplicates, leaving the IC completely unusable.

Therefore, a filter needs to be applied both ways. In Jira, this filter is applied in the WebHook specification in the system, where a JQL for outgoing messages can be specified. By specifying that any updates made by the IssueConnector (or the Api User, as described in appendix J) should be ignored, the IssueConnector will not receive these messages, and the loop will break.

This feature is not available in EasyAccess, which means that this filter must be implemented in the connected service. The EasyAccess SignalR client does not provide which user made the update, meaning that the same solution can not be used here. Therefore, the incoming message must be analyzed more carefully to determine if the update is in fact made in EasyAccess, or by the IC. The way this is done now is through a hack which reads the title of the message and screens it for which system it has originated from.

For every system to be connected, a way to stop this loop must be implemented. This is a well known problem for integration systems, and there is no universal way to solve it. It needs to be solved for each system individually.

5.3 Application Structure

The repository for the IssueConnector is available here[7]. The IssueConnector consists of several parts. At the moment, these are all part of the same solution, but that is no longer necessary, as the library has been published as a NuGet package. This section briefly describes each part of the application.

[7] https://github.com/oysteinhauan/IssueConnector

5.3.1 IssueConnectorLib

This is the class library that has been released as a NuGet package. It consists of standard object models that are used throughout the services, as well as providers for message buses, both publisher and consumer. It also includes some defined exceptions for the IssueConnector, to simplify the debugging and error handling. It also contains a database service, to connect to the database that contains user maps and project maps.

Required interfaces for succesfully creating a new service are also implemented here. A guide for creating a new service is available in appendix K.

5.3.2 IssueConnectorConfig

This is the configuration page displayed in appendix I. It is a basic webservice, that connects to the database of the IC and can insert new mappings. Only insert is supported at the moment, to prevent users from mistakenly deleting information. Here, the user can insert new project mappings, and define which Jira-url the project is registered at.

5.3.3 JiraService and EasyAccessService

These are the resource services for Jira and EasyAccess; instances of the model displayed in figure 5.5. These consist of the following folders:

- **Client.** This folder contains the object model for the client, implementing the interface IResourceClient. This is the object that is connected to the dedicated system.

- **Connector.** This is the connector, a static function that contacts the systems authentication service and receives an access token. This token is then stored in the **Client**.

- **Controller (only Jira).** This is the API that the WebHook in Jira connects to. It receives updates from the system, filters them, and sends them to the **Publisher**.

- **EventListener (only EasyAccess.)** EasyAccess uses SignalR for notifying events. This is the part of the application that listens for updates, and sends the filtered ones to the **Publisher**.

- **Filters.** The message filters of the service is located here. These filters evaluate the incoming messages, based on contents or headers, and decide what to do with them.

- **Mapper.** This folder contains the object mappers for the service. The messages are provided to these, and they map them to the corresponding object model in the dedicated system.

- **Models.** This folder contains all the pure object models that are specific to this service. These are strictly models, and contain no logic.
- **Services.** This folder contains the operating services of the service.
  - **Consumer.** The consumer's task is to subscribe to the messages coming from the broker, sending them through the filter, and handling them accordingly by sending them to the **CrudService**.
  - **Publisher.** The publisher receives outgoing messages, and publishes them to the broker.
  - **CrudService.** This is the service that interacts with the system, e.g. Jira, through the **Client**. The client has a RestClient, that is used to perform CREATE, READ, UPDATE, DELETE operations on the system, and use the **Mapper** to map the objects.

5.4 Pilot Projects

In spite of having a very short development time of only 3-4 months, the IC gathered a lot of attention from the industry. Several companies expressed their interest in testing the IC as soon as possible. Sweco, Rambøll and Betonmast Hæhre (hereby referred to as Hæhre) quickly followed through on their interest, and allocated some resources for setting up test projects. Both Sweco and Rambøll had their own Jira-installations that they were already using, while Hæhre had never used it before.

While Sweco showed great interest in testing the IC from several individuals, the testing process suddenly froze, and was unfortunately not revived in time for this thesis. This was due to the communication on their behalf suddenly went silent before all the required steps to connecting the IssueConnector were completed. Therefore, this pilot project is left out of this section.

5.4.1 Betonmast Hæhre

As described in chapter 4, Hæhre wants to start using Jira for their infrastructure projects. It was agreed that Trimble was going to host this initially, in order to simplify the connection between EasyAccess and Jira through the IC.

The project connected to Jira was a part of the E18 road in Norway. This project already existed in EasyAccess, with over 400 registered topics. These were exported to Jira, along with all comments and viewpoints, through a script. All the registered topics received a Jira-duplicate with matching values for every field. Once this script was done transferring issues, Jira was ready to be used, with established two way synchronization.

At the time of this thesis, this solution has been tested for about four weeks. Some feedback from Hæhre is available in appendix F. They are very happy with the way the IssueConnector synchronizes both systems in real time, so that both systems carry the same information at all times. Using the IssueConnector, Adrian

Saunders says "[...] we can combine different disciplines with different processes into one, common solution, in spite of using different software." This was the main objective of the IC, and it shows that the customer is realizing the benefits and potential of such an integration platform. Software and tools that may not be sufficient on their own, can become more powerful in seamless orchestration with others, without the need to develop another new system or platform.

Regarding potential for improvement specifically for the IssueConnector, primarily the process of connecting projects and users is brought up. There is a web service developed for exactly this, but it is a work-in-progress. The current solution is displayed in appendix I. The idea is that the administrator of each customer gets access to their own private page. Here they can add project mappings for the connected systems. At the moment, this is where the user maps are added as well.

It should be noted that the feedback received from Hæhre in general, through meetings, e-mails and conversations, has been overwhelmingly positive. It is widely agreed upon among the testers, that this solution has great potential in the infrastructure industry.

5.4.2 Rambøll

Rambøll[8] is a leading engineering, design and consultancy company, originally founded in Denmark in 1945. It employs more than 13.000 experts globally, and over 1500 in Norway. They were introduced to the IssueConnector, as one of their employees, Goran Huseinovic, is a participant in the SPP project. He expressed his interest in connecting their Jira to EasyAccess, and it was agreed that a test project should be set up.

Initially, the IssueConnector was connected to a project on their test environment installation of Jira. Following the guide in appendix J, we were able to configure their Jira to connect to the IC. There was a problem with the WebHooks, however, which meant that Rambøll unfortunately only were able to get synchronization one way, from EasyAccess to Jira. This is most likely because of SSL/HTTPS certificate issues, as the IssueConnector currently is deployed on HTTP protocol, while Rambøll's Jira is HTTPS. By the time we received the log files for Jira to debug this issue, there was no time to fix it before the due date of this project.

After testing for a few days on the test server, Rambøll decided they wanted to connect to their production server of Jira. One project in EasyAccess, Nittedal Stasjon, was connected to Jira, and all the required users were added to the user mapping database.

In spite of only having one-way synchronization, Rambølls feedback was also very positive. These are attached in appendix G. They were asked the same questions as Hæhre in appendix F. The feedback regarding the IssueConnector and Jira is similar to that of Hæhre. Instant issue creation in Jira upon Topic creation, enables instant visualization of the pending problem, and it is available for all project participants to see. By connecting the different systems to one common coopera-

[8]http://www.ramboll.co.uk/about-us

tive platform, generates opportunity to involve customers and stakeholders in the process, which provides them a continuous overview of the projects challenges.

Also Rambøll have only minor suggestions for improvement on the IssueConnector. The platform has been stable for this project as well, and issues in Jira are updated according to changes in EasyAccess. They have received a demonstration of how it works when it is synchronized both ways, and they are eager to solve the certificate problem to get this up and running full scale.

5.4.3 Results and Conclusions

The pilot projects were both short, as they were not configured until the middle of April 2018 and this project was due on May 31st. The feedback from this short testing period has been very positive, with only minor requests for improvement. The solution has proved to be quite stable, and has survived "pressure tests" when transferring several hundred topics at once through the IC. The participating testers are eager to continue using the IssueConnector, which is a great indicator of it's potential. Both projects have been surprisingly successful, given the IC's young age. There is an obvious interest for this software to be further developed, which should be a big motivator for Trimble and SPP.

5.5 Summary and Further Development

The IssueConnector is an integration platform, that is designed to enable communication between different tools in the infrastructure industry. It allows for instant synchronization between systems, meaning that updated only have to be manually entered in one system. It is designed in a way that makes it simple to add new services to the cluster, and thereby connecting new systems.

Developing the IssueConnector has been a very successful project. It gathered a lot of attention from the Norwegian infrastructure industry, and during the short duration of this project, we were able to set it up in three separate test projects, two of which were real, ongoing infrastructure projects at industry partners. The feedback from both audiences and testers has been overwhelming, and the application has managed to stay online for weeks on end without crashing, or becoming unavailable (with the exception of a power outage).

However, if the IC is heading for full release, there is a substantial amount of work that has to be done in order to ensure customer satisfaction. A handful important challenges and suggestions for future work are presented in appendix L. These involve proper deployment, database refactoring, encryption, clustering, certificate verification etc. If the IC is released and connected to several customers and different projects, it is vital that the customers data is protected and handled correctly.

Chapter 6

Conclusions and Future Work

The Norwegian infrastructure industry is affected by long lead times for development projects, and the process of planning and designing these projects is facing a shift in process methodology. There are many opportunities for improving use of digital tools to help streamline these processes, and introducing methods used in agile software development methodologies can have a great impact on the future of the industry.

By introducing tools used in agile software development projects, the industry can potentially gain better involvement between project participants and stakeholders. These tools also introduces the opportunity for the different disciplines to gain a better overview of the project, improving transparency and multi-disciplinary understanding. The utilization of these tools can be expanded through the use of systems integration, as the IssueConnector has demonstrated in this project. The fragmented infrastructure industry can potentially benefit greatly from a common tool for project and task management, that will aid in making the workflow more streamlined and efficient for the users.

6.1 Results of Research Questions

6.1.1 P1

The problem in question was exploring if tools designed for software engineering could be customized to satisfy the requirements from the Norwegian infrastructure industry. Jira was the selected tool for this project, an issue tracker and project management tool designed for software development projects.

By configuring Jira in collaboration with experts in the Norwegian infrastructure industry, and testing these configurations out in real projects, we were able to determine that Jira has a great potential in the industry. Several experts from different disciplines voiced their opinions, and the overall feedback was very positive.

An important condition for successfully implementing such software in the industry, is ensuring an intuitive user interface, and make the benefits of using the software obvious for the average end user. If the users see Jira as a helpful tool rather than an obstacle, the process of implementing it will become smoother.

It was broadly agreed upon, that Jira as a task management tool has great potential in the Norwegian infrastructure industry.

6.1.2 P2

Developing a platform for connecting already existing industry related software, as well as new tools like Jira, was the objective of the second part of this project. The outcome of this is the IssueConnector. It is a loosely coupled, distributed system, with an architecture based on the publish-subscribe design pattern. It synchronizes the connected systems in real time, through shared, standardized object models.

The success of this product is already apparent. The two pilot projects that have been conducted, generated very positive feedback, and participating companies are eager to continue using it. It provides a new level of integration between systems in the industry, and opens up doors for simplifying involvement between different disciplines. There is work that remains, regarding security, deployment, stability and everything mentioned in appendix L, but these are all obstacles manageable within a reasonable period of time.

6.2 Future Work

The industry still has a long way to go before the 50% time reduction objective is reached. The SPP project will conclude in 2019, and will hopefully establish new methodologies and processes for implementing concurrent engineering methods. A major challenge for the future of Jira, is introducing the system to all the users, and ensuring that the software actually is optimally utilized. The configurations that have been tested in this project need to be expanded, and introduced to a larger audience, to gain a better, less biased impression of its potential in the industry.

Should Jira be chosen as a common tool in the industry, a possibility is to reach out to the developer, Atlassian, and explain any shortcomings of the software. They are involved in their community, and answer feature requests and bug reports frequently and enthusiastically. If enough customers are on the fence because of only minor, necessary tweaks, there is a good chance they might look into it and propose a solution for other industries that software development.

Regarding the IssueConnector, the hands on work that needs to be done is listed in appendix L. It is very difficult to say how it eventually will turn out, but should the IssueConnector truly succeed in the market, a dedicated team to maintain and update the software is essential. The Norwegian infrastructure industry has clearly stated their desire for this product, meaning the future of the solution is highly dependent on the product quality upon release, and the maintenance of it.

Bibliography

[1] Niels Treldal, Hussain Parsianfar, and Jan Karlshøj. *Using BCF as a mediator for task management in building design*, pages 48–59. Rilem publications, 2016.

[2] CEITON Technologies. Diagram of point-to-point spaghetti structure. URL `http://ceiton.com/CMS/EN/workflow/integration-EAI.png`. Accessed: 22-03-2018.

[3] S. G. Rasmussen. *Integrated Concurrent Engineering i Samferdselsprosjekter*. NTNU, 2016.

[4] Fredrik Kirkevold. Sju av ti norske vei- og jernbaneprosjekter kraftig forsinket, 2013. URL `https://www.tv2.no/a/3986677/`. Accessed: 10-05-2018.

[5] Tor Sandberg. Derfor blir mange veier og jernbaner forsinket, 2018. URL `https://frifagbevegelse.no/njfmagasinet/derfor-blir-mange-veier-og-jernbaner-forsinket-6.158.532640.e4ad68a258`. Accessed: 10-05-2018.

[6] F. Olshausen and K. Dybvik. Bruk av concurrent engineering i reinertsen - hvordan gjennomføre concurrent engineering møter i reinertsens landbaserte prosjekter, 2015.

[7] Kommunal og moderniseringsdepartementet. *Lov om planlegging og byggesaksbehandling (plan- og bygningsloven)*. 2008. URL `https://lovdata.no/dokument/NL/lov/2008-06-27-71`.

[8] Alistair Cockburn et al. Manifesto for agile software development. URL `http://agilemanifesto.org/`.

[9] David Cohen, Mikael Lindvall, and Patrícia Costa. Agile software development. 01 2012.

[10] Nel Wognum and J Trienekens. *The system of Concurrent Engineering*. 01 2015. ISBN 978-3-319-13776-6.

[11] cPrime. What is agile? what is scrum? URL `https://www.cprime.com/resources/what-is-agile-what-is-scrum/`. Accessed: 16-05-2018.

[12] Ken Schwaber and Jeff Sutherland. The scrum guide - the definitive guide to scrum: The rules of the game, 2017.

[13] Leo Adell, 2013. URL http://www.belatrixsf.com/blog/benefits-pitfalls-of-using-scrum-software-development-methodology/. Accessed: 16-05-2018.

[14] Massimo Bandecchi. The esa concurrent design facility (cdf): Concurrent engineering applied to space mission assessments. pages 1–36, 01 2001.

[15] Yongwu Miao and Jörg M. Haake. Supporting concurrent design by integrating information sharing and activity synchronization. 1998.

[16] Youcef J-T. Zidane, Kjersti Bjørkeng Stordal, Agnar Johansen, and Susanne Van Raalte. Barriers and challenges in employing of concurrent engineering within the norwegian construction projects. *Procedia Economics and Finance*, 21(Supplement C):494 – 501, 2015. ISSN 2212-5671. 8th Nordic Conference on Construction Economics and Organization.

[17] Sigurd Wolden. *The Application of Concurrent Engineering in Infrastructure Projects*. NTNU, 2017.

[18] US National Bim Standard. Frequently asked questions - what is a bim? URL https://web.archive.org/web/20141016190503/http://www.nationalbimstandard.org/faq.php#faq1. Accessed: 21-05-2018.

[19] NBS. What is bim? URL https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim. Accessed: 21-05-2018.

[20] Rajat Agarwal, Shankar Chandrasekaran, and Mukund Sridhar. Imagining construction's digital future. 06 2016. URL https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imagining-constructions-digital-future.

[21] Aproplan. What is bim? what are its benefits to the construction industry? URL https://www.aproplan.com/blog/quality-management-plan-construction/what-is-bim-what-are-its-benefits-to-the-construction-industry. Accessed: 21-05-2018.

[22] BIMCollab. Ifc - and why should you use it?, . URL https://www.bimcollab.com/en/BIM/OpenBIM/IFC. Accessed: 15-05-2018.

[23] buildingSMART. What we do. URL https://www.buildingsmart.org/. Accessed: 14-05-2018.

[24] cobuilder. åpenbim standardene har blitt norsk standard. URL https://cobuilder.com/nb/aapenbim-standardene-har-blitt-norsk-standard/. Accessed: 15-05-2018.

[25] BIMCollab. Bcf - building collaboration format, . URL https://www.bimcollab.com/en/bim/openbim/bcf. Accessed: 15-05-2018.

[26] Dave Meyer. Organize your jira issues with subcomponents, 2013. URL https://www.atlassian.com/blog/jira-software/organize-jira-issues-subcomponents. Accessed: 10-05-2018.

[27] Peter Varhol. To agility and beyond: The history—and legacy—of agile development. URL https://techbeacon.com/agility-beyond-history%E2%80%94-legacy%E2%80%94-agile-development. Accessed: 21-05-2018.

[28] Microsoft. C# guide, 2018. URL https://docs.microsoft.com/en-us/dotnet/csharp/. Accessed: 09-05-2018.

[29] Microsoft. Dependency injection in asp.net core, 2016. URL https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-2.0. Accessed: 09-05-2018.

[30] Federico Silva Armas. Atlassian.net sdk. URL https://bitbucket.org/farmas/atlassian.net-sdk/wiki/Home. Accessed: 10-01-2018.

[31] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison Wesley, 2013.

[32] S.Roselin Mary and Dr Paul Rodrigues. Software architecture - evolution and evaluation. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 3(28):335–337, 2014.

[33] Techopedia. System integration (si), . URL https://www.techopedia.com/definition/9614/system-integration-si. Accessed: 09-05-2018.

[34] Ranjithprabhu K. and Sasirega D. Eliminating single point of failure and data loss in cloud computing. *International Journal of Science and Research (IJSR)*, 3(4):82–88, 2012.

[35] Troy Davis. What is a webhook? URL https://webhooks.pbworks.com/w/page/13385124/FrontPage. Accessed: 8-04-2018.

[36] Patrick Fletcher. Introduction to signalr, 2014. URL https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr. Accessed: 8-04-2018.

[37] Gregor Hohpe and Bobby Woolf. Message bus. URL http://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBus.html. Accessed: 8-04-2018.

[38] Mitre.org. Reliability, availability, and maintainability, 2017. URL https://www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/integrated-logistics-support/reliability-availability-and-maintainability. Accessed: 11-05-2018.

[39] GEIA 2008. Reliability program standard for systems design, development, and manufacturing. *SAE - GEIA-STD-0009*, 2008.

[40] Margaret Rouse. What is stateless? URL https://whatis.techtarget.com/definition/stateless. Accessed: 04-05-2018.

[41] Techopedia. Enterprise application integration, . URL https://www.techopedia.com/definition/1506/enterprise-application-integration-eai. Accessed: 22-03-2018.

[42] Microsoft. Generic interfaces (c# programming guide), 2015. URL https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/generics/generic-interfaces. Accessed: 22-04-2018.

[43] Chris Sevilleja. The ins and outs of token based authentication. URL https://scotch.io/tutorials/the-ins-and-outs-of-token-based-authentication. Accessed: 22-05-2018.

Appendix A

Feature Analysis

It should be noted that the number of software initially on this list was significantly larger, but many of them proved unsuitable for this project, and were therefore excluded from this paper. These included among others dapulse [1], StormBoard [2] and Realtime Board [3]. This was mainly due to the fact that they were more focused on planning and management, rather than issue tracking and logging.

[1] https://dapulse.com/
[2] https://stormboard.com/
[3] https://realtimeboard.com/

Technology analysis; SPP

Software	Weight, priority	JIRA, Confluence		Stormboard		TRELLO		BIMcollab	
Target Industry		Issue and project tracking				Issue tracking, planning			
			Score		Score		Score		Score
Flexible attributes	10		10		10		10	?	5
Flexible categorisation	10		10		10		10	?	5
Reporting, flexible listing	10	Y	10			Y	10		10
Flexible setup and Task status	10	Y	10		10	Y	10		7
Registry of participants	10		10		10		10		10
Cloud-based possibility on premises	10	Y	10		10		10		10
Exportation of log	10		10				10		10
Usage Complexity	10	Medium	7		10	Simple	10	Medium	7
Web support	9	Y	9		9	Y	9		9
Flexible process flow setup	9		9				9	?	4
Assign tasks	9	Y	9		9	Y	9	Y	9
REST API	9	Y	9		9	Y	9		9
Role based access control	8		8				8		0
Listing Templates	8		8		8		8		0
Planning/estimation , priorities	7	Y	7		7		7		0
Documentation Quality	7		7		5		7		4
Cost (See Notes)	7		5		5	cheap	7	medium	5
Relation mapping, modeling	6		4				6		0
Mobile support	6	Yes, WEB	4		6	APP	6		3
Touch screen support	5		0		5		5		0
Visual Planning	5	Y	0		5	Y	5		0
Notifications (in-app, sms, mail)	5		5		5		5		5
Setup Complexity	5	Medium	5		5		5	Simple	5
3D-model interface	4	N	0				0	Y	0
Kanban board	3	Y	3		2	Y	2		0
Total			**119**		**100**		**137**		**80**

Appendix B

Project Log by Erling Graarud

This is a document that presents a proposed project log. This log should containt all the relevant information to a project over the course of its lifecycle.

Project log
Ver 06.02.2017

Identifikatorer
Beslutninger
BIM

	Innmeldt		Tilknyttet område				Problemidentifikasjon				
ID	**Dato**	**Plassering**	**Fag**	**Tema**	**Geografisk**	**Descr**	**Avklaringsbehov**	**Krevet av / bestilt av**	**Bakgrunn/beskrivelse**	**ntatt beslutningstake**	**Interessenter**
1	9/12/2007		Vegteknikk		Veg 1	Beslutning	Asfaltert bredde veg?	TES			
2	2/20/2009		VA		Veg 2	Beslutning	Lukket drenering?				
3	12/11/2014		Elektro		Veg 1	Beslutning	Belysning krevet?				
4	6/3/2015		Geometri		Veg 1	Forutsetning	-		Vegklasse H1 valgt		
5	4/19/2016		Geometri		Veg 2	Aksjon	Dimensjonerende hastighet?	EGR	Må avklares	Vegdirektoratet	Vegkontoret
6	11/24/2016		Kostnad; Bru		Dalen	Aksjon	Kostnad for bru 1?	NNN	Behov for sammenlikning bru mot kulvert		
7	2/6/2017		Bru		Dalen	Vedtak					

Punkt	*Drift/Vedlikeholc Stikkord*	*Informasjon*
Linje	*Anleggsgjennomføring*	*Aksjon*
Objekt	*Geologi*	*Forutsetning*
Område/areal		*Beslutning*
Volum		*Antakelse*
"Blank"		*Vedtak*
Hele prosjektet?		

Kobling til BIM — Mulig å legge inr Frie stikkord? — Nedtrekksliste? — Nedtrekksliste?
Tegneregler? — - kolonseparert? — Fra interessentanaly: — Fra interessentanal

Merknader / problemområder:

Bruks-algoritmer - Hvordan behandle utvikling og historikk?
Spørre-apper
Tegneregler for visning av resultater, - ligger i BIM
Redigeringsverktøy, sette inn nye kolonner mm
Bør være fleksibelt
Må kunne eksporteres til Excel og dxw
Grensesnitt

Beslutningsprosess					Beslutning					Modellstatus			
Frist	**Ansvarlig**	**Åpen**	**Overdue**	**Lukket**	**Valg**	**Begrunnelse**	**Beslutningstaker**	**Beslutningsform**	**Dato**	**Planfase**	**Status**	**Revisjon**	**Modenhet**
		X			10,0m x 2	Ihht gammel håndbok	Vegdirektoratet	Avklaringsnotat	3/10/2008	KDP	Plan i arbeid		
					Nei				3/12/2009	KDP	Plan i arbeid		
		X			Ja				2/4/2015	Detalj/Regulering	Plan i arbeid		
				X	Vegklasse H1 for Veg1		Prosjektet		8/11/2015	Detalj/Regulering	Plan i arbeid		
8/10/2016		X	X										
12/1/2016		X											
				X	Form og plassering vedtatt		Kommunen	Vedtatt reguleringsplan	1/1/2017	Detalj/Regulering	Vedtatt plan	0	Nivå 2

SP-møte nr x	*SP-sesjon*	*Utredning*	*Plan i arbeid*	*Løpenr*	*Nivå 1*
NN	*Avklaringsnotat*	*KDP*	*Høringsutgave plan*		*Nivå 2*
	Off. godkjenning	*Detalj/Regulering*	*Vedtatt plan*		*Nivå 3*
	Brev	*Byggeplan*	*Under utførelse*		
		Konkurransegrunnlag	*Bygget*		
		Byggefase			
		FDV-dokumentasjon			
		Driftsfase			
		Vedlikeholdstiltak			

Nedtrekksliste?

yse

Appendix C

SPP Application

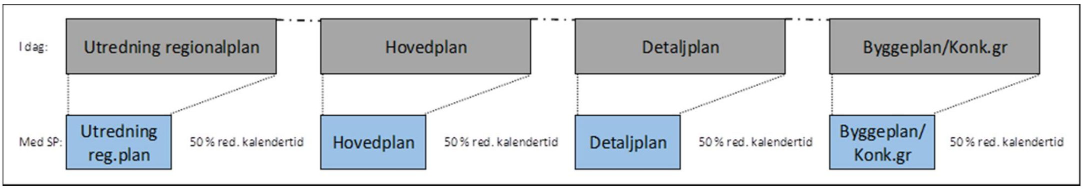
Prosjektbeskrivelse

DEL 1: Innovasjonen

1. Overordnet idé

Innen samferdsel er det i dag stort fokus både politisk og på etatsnivå for å senke planleggingstiden betydelig. På politisk hold er det uttalt at planleggingstiden skal halveres. Dette er også høyt prioritert hos de store byggherrene innen samferdsel. Hittil er ikke denne utfordringen løst. Noe av grunnen til dette er at det kreves samarbeid og store ressurser på tvers av fag og formelle roller. Oppgaven vil være for stor for en enkelt aktør med hensyn på ressurser, kompetanse og koordinering.

Vi har nå satt sammen en faglig bred og kompetent gruppe fra ulike aktører i næringen for å ta tak i denne problemstillingen. Prosjektets overordnede ide er å utvikle en metodikk for å redusere plan- og prosjekteringstiden innen samferdselsprosjekter med minimum 50% kalendertid. Dette vil spare det norske samfunn for svært store summer gjennom tidligere realisering av den samfunns-økonomiske gevinsten. Det er også et stort potensiale i den offentlige planbehandlingstiden i tillegg til områdene vi forbedrer ved FoU-prosjektet, men vi har avgrenset FoU-oppgaven til plan- og prosjekteringsutførelsen.



Figur 1: Potensiale for tidsreduksjon

FoU-prosjektet skal derfor:

- Utvikle ny plan-/prosjekterings-metodikk (heretter kalt PPM)
- Utvikle nye tilhørende IKT-verktøy
- Implementere metodikken i partnerbedriftene og samferdselsbransjen nasjonalt

Prosjektet har to hovedfokusområder; forske frem etablering av nye systemer for PPM og etablering av nye IKT-verktøy.

1. Metodikk (PPM)

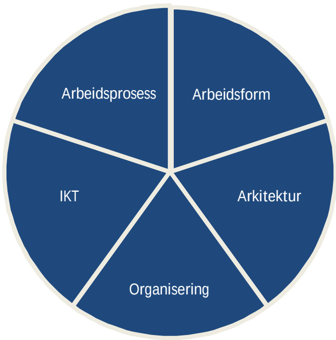
Samferdselsbransjen har siste 5-10 år gradvis påbegynt en endringsprosess gjennom innføring av 3D-modellering/BIM (Building Information Modeling) av prosjektene.

Til tross for denne teknologiske forbedring har samferdselsbransjen ikke klart å realisere det store potensiale for effektivitetsforbedring som plan- og –prosjekteringsprosessen har. Nå ser vi det påkrevet for oss selv og samfunnet og ta ut dette potensialet. Den vesentligste faktoren for slik utnyttelse er bedre samhandling og parallellarbeid. Målet er derfor utvikling av «Samtidig Prosjektering» (heretter kalt SP) i samferdselsprosjekter, som er tittelen på denne søknaden.

Prosjektet skal utvikle en arbeidsform blant annet basert på erfaringene fra andre bransjer (f.eks. ICE, VDC, Concurrent Design, Lean og Scrum) og mulighetene som BIM gir, men som samtidig er spesielt tilpasset samferdselsprosjekter.

2. IKT-verktøy
Gevinsten av SP er avhengig av at IKT-verktøyene er tilpasset for å støtte opp om behovene. Det er derfor lagt inn vesentlige poster i prosjektbudsjettet for utvikling og tilpasning av partenes egne systemer: Novapoint og Epsis TeamBox.

Utover ovennevnte hovedfokusområder er det også andre suksesskriterier som må være tilstede for å danne helheten i konseptet. Dette omfatter i hovedsak prosjektorganiseringen, utformingen av prosjektarealene og tilpasning av arbeidsprosesser.



Figur 2: SP -suksesskriterier

Prosjektet vil fokusere på alle disse fem suksesskriteriene gjennom FoU-prosjektet.

Basis for SP ligger i fremskaffelse av faglig riktige og rettidige beslutninger i et optimalt og planlagt system. Beslutningstakerne skal ha nødvendig og forståelig informasjon (og/eller personer med fagekspertise) tilgjengelig når beslutning skal tas.
SP og samhandling med digitale verktøy effektiviserer planprosessen, der gevinsten bl.a. ligger i å avdekke faktiske forhold og konflikter tidligst mulig, samt bevisstgjøre plandeltakerne om dette. En slik beslutningsform anses best gjennomført ved spesielt planlagte og sammensatte arbeidsmøter, i det etterfølgende kalt «sesjoner».

2. Innovasjonsgrad
I hovedtrekk omfatter innovasjonen følgende elementer:

Tjenester/Leveranser:
- Ny bransjestandard for plan- og prosjekteringsgjennomføring i Norge
- Ny PPM (plan- og prosjekteringsmetodikk) – for planleggingsfirmaene i gruppen
- Salg av prosessledelse og metodikk
- Håndbøker og veiledere i samspill med de offentlige byggherrene

For partnerne vil dette gi nasjonalt og internasjonalt ledende markedsposisjon for utførelse av større samferdselsanlegg. Både ved egen utnyttelse av metoden samt ved salg av prosesstyring/bistand til eksterne prosjekter og partnere. Samtidig vil metoden gi mer effektiv ressursutnyttelse av fagpersonell og mulighet for å utføre flere og større prosjekter.

For samfunnet vil utnyttelsen av en slik metodikk gi svært store besparelser i tid og kostnader.
For full effekt vil det være viktig at dette blir bransjestandard, ved utarbeidelse av nødvendige håndbøker og veiledere.

Produkter:

- Integrert samhandlingsverktøy og beslutningsstøtteverktøy, ved å gå fra «prosjekteringsverktøy» til «beslutningsverktøy»
- Nye applikasjoner og tilpasninger til samhandlingsverktøyet, herunder:
  - GIS- data (kulturminner, verneområder m.m.), risiko, kostnad, mengder
  - Håndtering av miljøbudsjett
  - Håndtere tidsdimensjoner (anleggsgjennomføring mm.)
  - Innarbeide trafikkberegninger
  - Samvirke mellom Novapoint og Epsis TeamBox

For partnerne vil dette være salgbare funksjoner/moduler som ikke eksisterer i dag, samt at verktøyene vil gi klare konkurransefortrinn for de av partnerne som selger tjenester basert på disse verktøyene. Novapoint og Epsis Teambox skal bli et nasjonalt og internasjonalt ledende produkt for denne metodikken.

For samfunnet vil disse verktøyene være sentrale løsninger for å oppnå de politiske og etatsmessige målene om effektivisering av planprosessene.

Organisering og forretningsmodeller:

- Nye samspillsformer mellom fagansvarlige og beslutningstakere ved samtidig involvering i prosessen – i såkalte sesjoner
- Nye kontraktsformer for å nyttiggjøre ovenstående tjenester og produkter (eksempelvis «Best Value»-prinsippet)
- Økt fortjeneste ved nye former for tjenestesalg, eksempelvis arbeid etter fastpris kontra utførte timer

For partnerne vil dette primært gi effekt i høyere kvalitet og involvering av alle fag og medarbeidere i samspill med byggherre.
For samfunnet vil effektiv utnyttelse av ressurser og tidsbruk gi direkte besparelser på investeringsprosjekter.

3. Verdiskapingspotensial

Partner	Verdiskapning	Hvordan	Antatt verdi av FoU prosj.
VNPT	Økt konkurranseevne Effektiv gjennomføring Sterkere markedsposisjon	SP vil bidra til at VNPT får et betydelig konkurransefortrinn i konkurransen om fremtidige samferdselsprosjekter. Dette på grunn av økt kvalitet i prosjektering og redusert gjennomløpstid. Selv om dette konkurransefortrinnet ikke vil vedvare over tid, vil prosjektet bidra til å befeste VNPT sin posisjon som et proaktivt og innovativt selskap.	Vi antar 10% omsetningsøkning og 5% resultatgradøkning pga forbedrede prosesser. Omsetningen har ligget på ca 120 MNOK de siste årene. Dette gir en verdi på: 120*1,1*0.05 = 6,6 MNOK/år. Antar effekt i 5 år, dvs en total gevinst på **33MNOK**
Rambøll	Som for VNPT over	Som for VNPT over. I tillegg er Rambøll et globalt selskap med kontorer i alle	Prinsipp som VNPT. Omsetningen innenfor transport i Rambøll har ligget på ca 520 MNOK de siste

		verdensdeler. SP vil gi Rambøll en Global markedsposisjon over sikt.	årene. Dette gir en verdi på: 520*1,1*0.05 = 28,6 MNOK/år. Antar effekt i 5 år, dvs en total gevinst på **143 MNOK**
Metier	Økt omsetning Nye markeder	Prosjektet vil sette standarden for hvordan fremtidige samferdselsprosjekter bør gjennomføres. For Metier innebærer dette nye muligheter for utrulling av metodikken til andre aktører, fasilitering, kurs/foredrag og relaterte forbedringsprosjekter	Antar et snitt på 5 nye oppdrag pr år (fasilitering av metode, prosjektledelse for byggherrer, kurs). Antatt årlig total omsetningsøkning på 10 MNOK og 1,5 NOK i gevinst. Samlet verdi over 5 år er 5 * 1,5 = **7,5 MNOK**
Epsis	Økt omsetning Nye markeder	Tilpasning av Epsis TeamBox til samferdselsprosjekter åpner et nytt og spennende marked for programvaren og mulighet til økt omsetning. I likhet med Metier innebærer dette nye muligheter for utrulling av metodikken til andre aktører, fasilitering, kurs/foredrag.	Estimerer årlig forbedring av driftsresultat på 2 MNOK basert på nye markedsandeler innen samferdselssegmentet. Over 5 år, gir dette en gevinst på **10 MNOK**.
VNS	Økt omsetning Globalt produkt	Prosjektet vil bidra til en ny versjon av Novapoint, og vil befeste programvarens posisjon og attraktivitet. Dette vil igjen føre til økt omsetning	Antar 10% omsetningsøkning uten ekstra kostnader. Antar at effekten holder seg i 5 år. Årsomsetningen er på ca 100 MNOK. Dette gir en gevinst på 100*0.1*5 = **50 MNOK**
NTNU	Rekruttering Renommé	Prosjektet vil bidra til ytterligere utvikling av NTNU sin kompetanse innen prosjekteringsledelse, lean-prinsipper og tidligfasen i offentlige prosjekter, og tiltrekke enda flere studenter og kursdeltakere	Anses ikke relevant å tenke overskudd/gevinster her
JBV	Redusert lengde på planprosesser Reduserte kostnader Høyere kvalitet på planer	Prosjektet vil bidra til redusert gjennomløpstid og tidligere realisering av fremtidige jernbaneprosjekter. Samtidig vil kvalitetsøkningen i prosjektene føre til lavere prosjektkost og et bedre sluttprodukt for brukerne, og de ansvarlige for drift og forvaltning	SP antas ville redusere tilleggsordre/feil, samt redusere prosjekteringstid. Mulig besparelse vurderes her til 10% av estimert total-kostnad for InterCity (IC) på 100 mrd. For IC gir dette en gevinst på 100*0,1 = **10 MRD**

4. Forskningsbehovet

Hovedutfordringene dette prosjektet skal adressere er at fasene med planlegging og prosjektering av infrastrukturprosjekter har liten grad av integrasjon på tvers av de mange deltakende aktørene, arbeidet er beheftet med feil og mangler og det tar generelt for lang tid. Dette er ikke en ny erkjennelse, men har vært kjente problemer lenge. Imidlertid er vi nå kommet til et punkt der kjente prinsipper for å bøte på situasjonen er i ferd med å kunne støttes av IKT-verktøy av tilstrekkelig modenhet, og målet med dette prosjektet er å realisere dette potensialet.

Dette gir et todelt forskningsbehov:

1. Ett behov har opphav i det faktum at prinsippene for såkalt samtidig prosjektering (SP) (på engelsk *concurrent engineering*, ofte forkortet CE) er velkjente, **men ikke like bredt anvendt i relevante bransjer**. Til tross for målbare, høye gevinster fra bruk av samtidighet i pionerbransjer, har en rekke andre bransjer, inkludert infrastrukturprosjekter, ikke evnet å ta dette i bruk. Dette skyldes nok hverken at prinsipper og metoder for samtidighet ikke er relevante eller at de ikke lar seg implementere i ulike bransjer, men at det er gjort for lite virksomhetsnær forskning rundt tilpasning og implementering av prinsippene
2. Den andre årsaken til lav utbredelse og som er det andre primære forskningsbehovet er at det ikke finnes nødvendige IKT-verktøy som er tilpasset ulike bransjer og typer prosjekter.

Det kreves derfor anvendt og innovasjonsrettet forskning som både løser utfordringene ved "oversettelse" av SP til samferdselsprosjekter, utvikle teknologiske løsninger og verktøy samt demonstrerer at det "virker" og hvilke effekter det gir. Som man ser av nettopp manglende implementering skjer ikke dette "av seg selv". Hvert prosjekt har stramme tidsfrister, krav til gjennomføringsmodeller, anskaffelsesmodeller som legger sterke føringer, osv. som gjør at aktørene ikke tør eller kan eksperimentere med nye metoder som SP.

Angående SP så gjorde Smith (1997) en undersøkelse av historien til CE og fant at noen av disse prinsippene var kjent før 2. verdenskrig og ble aktivt anvendt under krigen for å møte tidspress for å ta frem ny teknologi og våpensystemer. Etter krigen ble metodene lagt bort og gjenoppstod ikke før sent 1970-tallet/tidlig 1980-tallet. Fra da av har dette vært et tema med relativt mye forskning på ulike måter å oppnå raskere og bedre utviklingsprosesser (for produkter, prosjekter, osv.). I 1997 konkluderte Smith med at denne kunnskapen ikke i tilsvarende grad har vært tatt i bruk i næringslivet. Det er vanskelig å finne konkrete data på dette, men det synes å være store bransjeforskjeller i utbredelse siden den gang, med nokså utstrakt bruk i tradisjonell produktutvikling, også høy grad av bruk i teknologisk avanserte bransjer med énstykk- eller lavvolum-produksjon, f.eks. luft- og romfart, maritimt utstyr, osv., men adskillig mindre utbredelse i rene prosjekter i bransjer som olje og gass, bygg og anlegg, infrastruktur, osv. (IKT-prosjekter er et unntak der samtidighet oppnås gjennom ulike *agile* metoder).

Det finnes mer teoretisk forskning som forsøker å øke effektiviteten i prosjektering, under betegnelser som "lean engineering" (på norsk gjerne oversatt til *trimmet prosjektering*) som et resultat av tilpasning av "lean manufacturing" til en prosjektkontekst under overskriften "lean construction" (Koskela, 1992). I en videreføring av dette arbeidet utviklet Ballard (2000) metoden "lean project delivery" som også inkorporerer "lean design". Både Koskela, Ballard og andre forskere har demonstrert bruken av disse prinsippene i caseprosjekter, men da primært i tradisjonelle og også mer avanserte bygg, for eksempel sykehus (se for eksempel Ballard (2008) fra et amerikansk sykehus og Andersen, Belay og Seim (2012) fra anvendelse ved byggingen av St. Olavs Hospital i Trondheim). Michael J. Wodalski, Benjamin P. Thompson, Gary Whited, Awad S. Hanna (2011) gjennomførte en studie for å se på fordeler ved bruk av lean-prinsipper i samferdselsprosjekter og barrierer mot å ta slike i bruk. De fant at en rekke prinsipper og verktøy har stort potensiale for å gi forbedringer (f.eks. "involverende planlegging" (collaborative planning), avklaringsmøter før en oppgave skal starte, simulering og modellering for å støtte virtuell planlegging, inkludert bruk av BIM, osv.). De fant også spesielle forhold ved denne typen prosjekter som vanskeliggjør bruk av disse prinsippene, som at offentlige prosjekter setter strenge føringer for kontrahering av leverandører gjennom anbudskonkurranser, sterke tradisjoner for sekvensielle kontrakter der ulike aktører har ansvar for henholdsvis overordnet design, prosjektering for ulike fag og bygging, mangel på kultur for samarbeid og samspill på tvers av aktører og fag.

I de senere årene har det blitt mer vanlig å prosjektere avanserte samferdselsprosjekt med 3D-modeller i tillegg til (og etter hvert erstatte) 2D-tegninger. Denne utviklingen er bl.a. støttet opp av Statens vegvesen og Jernbaneverkets utgivelse av veiledere for modellbasert prosjektering. Det finnes i dag flere ulike IKT-løsninger som benyttes for prosjektering av samferdselsanlegg. Disse systemene fungerer som selvstendige program og prosjekteringsresultatene er ofte utfordrende å integrere i en felles modell. Den vanligste måten å løse dette på i dag er filutveksling og sammenstilling i egne visualiseringsprogram som f.eks. Navisworks og Virtual Map. For å få tatt ut det fulle potensialet i SP, er det viktig at de prosjekterte løsningene blir hurtig integrert, visualisert og dokumentert i en felles 3D-modell. For at arbeidssesjonene i SP-prosessen skal fungere godt, må endringer som gjøres i designet av de ulike aktørene umiddelbart kunne vises i den integrerte modellen. Effektive arbeidssesjoner i SP-prosessen vil også kreve bruk av informasjonselementer fra flere ulike datakilder i tillegg til 3D-modellen. Alle disse kildene må kunne deles og integreres i en felles arbeidsflate for bruk i sesjonene uten involvering av tung IT-kompetanse, og dette FoU-prosjektet skal utvikle en programvare for dette formålet.

Dette foreslåtte FoU-prosjektet ligger forøvrig faglig sett opptil andre pågående prosjekter med BIA-støtte, men uten direkte overlapp. Dette gir rom for å høste samordningsgevinster mellom prosjektene:

- **SpeedUp** (prosjekteier Reinertsen), har som mål å redusere varigheten av både tidlig- og gjennomføringsfasen av byggeprosjekter.
- **SamBIM -** Samhandling i byggeprosesser, med BIM som katalysator (prosjekteier Skanska), er også et meget relevant prosjekt som har som mål å utvikle og etablere prosesser og samhandlingsmodeller understøttet av BIM får å øke verdiskapningen i byggeprosjekter.
- **Integrert metodikk for prosjekteringsledelse** (prosjekteier Veidekke), har som mål å utvikle en ny metodikk for prosjekteringsledelse.

DEL 2: FoU-aktivitetene

5. Mål

Hovedmål:

Etablere ny Plan- og Prosjekterings-metodikk (PPM) med tilhørende verktøy

Delmål (realisering/milepæler):

R1 Endelig målformulering og prosjektplan
R2 Prosjektstyringsbasis etablert
R3 Spredningsplan etablert
R4 Forskningsmetodikk etablert
R5 Pilotprosjekt Byggeplan-SP avsluttet
R6 Forsknings- og erfaringsdokumentasjon kartlagt
R7 Metodemanual ver. 0 etablert
R8 Gjennomføringsplan for test Byggeplan-SP
R9 Gjennomføringsplan for test KDP-SP
R10 Pilotprosjekt Forstudie til KDP avsluttet
R11 30% av sesjonene i testprosjekt Byggeplan gjennomført
R12 30% av sesjonene i testprosjekt KDP gjennomført

R13 60% av sesjonene i testprosjekt Byggeplan gjennomført
R14 60% av sesjonene i testprosjekt KDP gjennomført
R15 Testprosjekt Byggeplan-SP avsluttet
R16 Testprosjekt KDP-SP avsluttet
R17 Forsknings- og effektdokumentasjon fra testprosjekter etablert
R18 Metodemanual ver. 1 etablert
R19 Metodeverk for SP implementert i deltakerbedriftene

Resultater/Effekter:

- Redusert kalendertid med 50% for planlegging / prosjektering pr planfase.
- Prosjektet skal utvikle parametere for å måle kvalitetsendring som følge av metoden. Prosjektet skal oppnå 25-30% bedring av kvalitet
- Færre feil og mangler – reduksjon av tilleggsarbeider
- Riktigere kostnadsoverslag
- Effektiv og tidlig problemidentifikasjon for alle fagfelt. Sikre at alle relevante problemstillinger avdekkes i første fjerdedel av planfasen
- Redusere antall revisjoner av plangrunnlag i de enkelte faser

6. FoU-utfordring og -metode

Som beskrevet under punktet om forskningsbehov er dette prosjektet bygget på to forhold:

- Metodikk (PPM)
- IKT-verktøy

Den sentrale problemstillingen i prosjektet er å oppnå kortere varighet av plan- og prosjekterings-fasene i store samferdselsprosjekter gjennom SP basert på nye samhandlingsmetoder og videreutviklede IKT-verktøy. Tolkes denne problemstillingen vidt kan det gi opphav til et altfor omfattende forskningsprosjekt. Vi avgrenser det derfor til følgende:

- I motsetning til mesteparten av tidligere forskning på SP, som har vært innrettet mot enten olje- og gassprosjekter og byggeprosjekter, setter vi dette prosjektet i en kontekst av store samferdselsprosjekter. Det finnes også mye forskning som viser at problemer med forsinkelser og kostnadsoverskridelser i store prosjekter er størst i samferdselsprosjekter, for eksempel Love et al (2014), som igjen viser til de mest kjente tidligere studiene, og Welde et al (2014), som så på norske prosjekter
- Det ligger ikke innenfor prosjektets omfang å gjøre mer grunnleggende forskning på ”concurrent engineering” eller avledede konsepter, men tilpasning og anvendelse av disse i denne konteksten. Love et al (2012) gjennomførte casestudier av to caseprosjekter (benevnt som infrastrukturprosjekter) med utgangspunkt i å finne årsaker til prosjekteringsfeil og løsninger for å unngå slike feil. Konklusjonen var at årsakene i stor grad var velkjente og at å unngå feil var et spørsmål om å ta i bruk kjente teknikker på en helhetlig måte (f.eks. bedre rutiner, ressurshåndtering, teknologi som BIM, fasegjennomganger, kontraktsmessige incentiver, osv.). I en annen studie så Tribelsky og Sacks (2011) på informasjonsflyt mellom ulike fag i prosjekteringen av infrastrukturprosjekter. De fant en positiv korrelasjon mellom prosjektsuksess (i form av overholdelse av tid og kost) og kvaliteten på informasjonsflyten mellom aktørene i prosjektet, fasilitert av teknikker fra Lean prosjektering
- Derimot er det en klar ambisjon om å anlegge FoU-arbeidet så nært opp til praksis at det kan gjennomføres konkret utprøving av nye løsninger i reelle caseprosjekter

Avgrensningen over viser til noen få nyere studier om søknadens tema. Både som et ledd i pågående arbeid med utvikling av relaterte undervisningstilbud ved NTNU og som en del av arbeidet med denne søknaden er det gjennomført omfattende litteratursøk i relevante tidsskrifter, rapportserier og bøker gjennom databaser tilgjengelige via universitetsbiblioteket ved NTNU.

Med hensyn til metodevalg vil dette prosjektet kombinere flere forskningsmetoder. På et overordnet nivå vil forskningsdesignet bestå av en kombinasjon av teknologiutvikling, aksjonsforskning og casestudier:

- Teknologiutviklingen vil følge anerkjente prinsipper for teknologi- og produktutvikling gjennom å kartlegge behov og utarbeide og prioritere relevante brukerscenarier, kartlegge eksisterende løsninger, utforme løsningskonsepter basert på prioriterte brukerscenarier, utvikle verktøyene gjennom en iterativ prosess med korte iterasjoner og brukerevaluering ved avslutningen av hver iterasjon (Scrum) og teste ut løsningene fortløpende i de pågående prosjektene
- Aksjonsforskning defineres på ulike måter og med varierende grad av ”radikalisme”, men grunnprinsippet er at forskningsresultater skapes i et fellesskap mellom forskere, utviklingsmiljøer og anvendelsesaktører (i motsetning til at forskere bruker anvendelsesaktører som rene laboratorier en studerer utenfra). Med ambisjoner om å drive praksisnær forskning, vil dette prosjektet måtte utformes slik at resultatene skapes i felleskap mellom disse aktørene
- Casestudier er en klassisk metode der det ikke er mulig å gjennomføre innsamling av store mengder ”nøytrale” data om fenomenet det forskes på, men der man i stedet gjør mer detaljerte og rikere studier i et fåtall case som grunnlag for utvikling av ny kunnskap og nye løsninger som forhåpentligvis kan generaliseres og overføres til andre kontekster

Under dette overordnede forskningsdesignet vil det anvendes en rekke mer konkrete forskningsmetoder og teknikker for innsamling og analyse av data:

- Bruk av prosjektdata fra caseprosjektene om relevante forhold, både bakgrunnsinformasjon og data om prosjektaktiviteter, typisk om informasjonsflyt (type informasjon, mellom hvilke aktører, med hvilken frekvens, osv.), verktøybruk, kvalitet på prosjekteringsgrunnlag, osv,
- Observasjoner av bruk av eksisterende og nye løsninger/verktøy og hvordan dette påvirker samhandling, prosjekteringsgrunnlag, BIM-modeller, osv.
- Intervjuer med deltakere i caseprosjektene for å kartlegge behov, erfaringer med nye løsninger, osv.
- Effektmåling av nye løsninger, gjennom innhenting av data om tid, kostnader, kvalitet på dokumentasjon, feil, samarbeidsklima, osv.

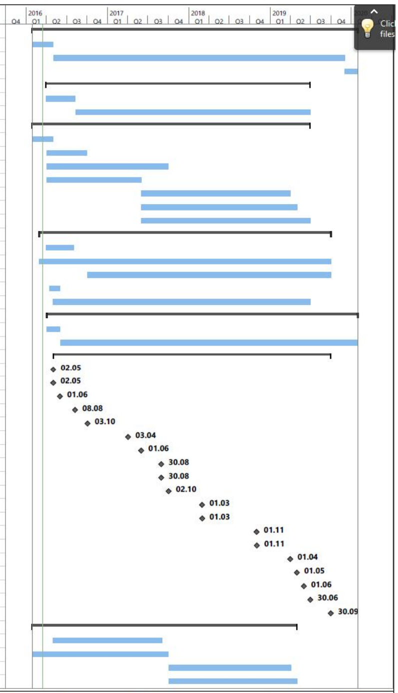
Vi gjør oppmerksom på at disse metodiske grepene er de vanligst anvendte i forskning på ”sosiale fenomener” og prosesser i virksomheter. I dette prosjektet representerer de hovedsakelig kilder til data om effektene av ny teknologi/nye løsninger. Det primære målet er som nevnt å utvikle ny teknologi som aktivt tas i bruk i caseprosjektene, og denne typen ”ingeniørmessig” forskning finnes det få omforente metodebegreper for.

De faglige målene med prosjektet er ambisiøse. Det er en grunn til at SP i liten grad er implementert i store infrastrukturprosjekter i dag; dette krever en kombinasjon av solid teoretisk forståelse for prinsippene for samtidighet og utvikling av helt nødvendige IKT-verktøy for å tillate bruk av disse prinsippene. Vi mener vi har et konsortium av aktører som daglig lever på forskningsfronten og er i stand til å håndtere disse utfordringene.

7. Prosjektplan

a) Hovedaktiviteter og milepæler i prosjektet. I nedenstående diagram er hovedaktiviteter og milepæler i FoU-prosjektet vist:

ID	Task Name	Start	Finish	Ansvarlig	Deltakende partner
1	**H1 Prosjektledelse og styring**	**Fri 29.01.16**	**Wed 29.01.20**	**Metier**	
2	Etablere styringsunderlag	Fri 29.01.16	Fri 29.04.16		MET, VNPT
3	Oppfølging og rapportering	Wed 04.05.16	Fri 29.11.19		MET, VNPT
4	Administrativ avslutning	Mon 02.12.19	Wed 29.01.20		MET, VNPT
5	**H2 Forskning**	**Tue 29.03.16**	**Fri 28.06.19**	**NTNU**	
6	Etablere forskningsmetodikk	Tue 29.03.16	Mon 08.08.16		NTNU, VNPT, EPS
7	Evaluere effekter	Wed 10.08.16	Fri 28.06.19		NTNU, VNPT, EPS
8	**H3 Metodeutvikling**	**Fri 29.01.16**	**Fri 28.06.19**	**VNPT**	
9	Plan for metodeutarbeidelse	Fri 29.01.16	Fri 29.04.16		VNPT, RAM, VNS
10	Pilotprosjekt - Erfarignsinnhenting. Byggeplan	Mon 04.04.16	Fri 30.09.16		VNPT, VNS, RAM
11	Pilotprosjekt - Erfaringsinnhenting. Forstudie til KDP	Mon 04.04.16	Fri 29.09.17		VNPT, VNS, RAM
12	Etablere metode og dokumentasjon til håndbok ver 0	Mon 04.04.16	Tue 30.05.17		Alle
13	Test-prosjekt -Gjennomføring av metodikk i fullskala. Byggeplan	Thu 01.06.17	Mon 01.04.19		VNPT, VNS, RAM
14	Test-prosjekt - gjennomføring av metodikk i fullskala. Kommunedelplan	Thu 01.06.17	Wed 01.05.19		VNPT, VNS, RAM
15	Optimalisering av metode og dokumentasjon til håndbok ver 1	Thu 01.06.17	Fri 28.06.19		Alle
16	**H4 Verktøyutvikling**	**Tue 01.03.16**	**Mon 30.09.19**	**VNS**	
17	Oppstartsaktiviteter mhp maskinvareoppsett, serverløsning, programvare mm	Fri 01.04.16	Mon 01.08.16		VNS, EPS
18	Tilpassing/viderutvikling av eksisterende programvare	Tue 01.03.16	Mon 30.09.19		VNS, VNPT, NTNU,EPS, RAM
19	Ny programvarefunksjonalitet (f.eks 4D, 5D, risiko, forbedret visualisering mm)	Mon 03.10.16	Mon 30.09.19		VNS, VNPT, NTNU,EPS, RAM
20	Avdekke behov for videreutvikling og tilpasninger av Epsis TeamBox	Fri 15.04.16	Wed 01.06.16		EPS
21	Videreutvikle og tilpasse Epsis TeamBox for bruk innen samtidig prosjektering	Mon 02.05.16	Fri 28.06.19		EPS
22	**H5 Formidling og spredning**	**Mon 04.04.16**	**Wed 29.01.20**	**Metier**	
23	Etablere formidlingsplan	Mon 04.04.16	Wed 01.06.16		MET, VNS, VNPT, RAM, EPS, NTNU
24	Følge opp tiltak i formidlingsplan (detaljeres ytterligere når planen er etablert)	Fri 03.06.16	Wed 29.01.20		MET, VNS, VNPT, RAM, EPS, NTNU
25	**SENTRALE MILEPÆLER FOR FoU-AKTIVITETENE**	**Mon 02.05.16**	**Mon 30.09.19**		
26	R1 Endelig målformulering og prosjektplan	Mon 02.05.16	Mon 02.05.16		
27	R2 Prosjektstyrignsbasis etablert	Mon 02.05.16	Mon 02.05.16		
28	R3 Spredningsplan etablert	Wed 01.06.16	Wed 01.06.16		
29	R4 Forskningsmetodikk etablert	Mon 08.08.16	Mon 08.08.16		
30	R5 Pilotprosjekt Byggeplan-SP avsluttet	Mon 03.10.16	Mon 03.10.16		
31	R6 Forsknings- og erfaringsdokumentasjon kartlagt	Mon 03.04.17	Mon 03.04.17		
32	R7 Metodemanual ver 0 etablert	Thu 01.06.17	Thu 01.06.17		
33	R8 Gjennomføringsplan for test Byggeplan-SP	Wed 30.08.17	Wed 30.08.17		
34	R9 Gjennomføringsplan for test KDP-SP	Wed 30.08.17	Wed 30.08.17		
35	R10 Pilotprosjekt Forstudie til KDP avsluttet	Mon 02.10.17	Mon 02.10.17		
36	R11 30% av sesjonene I testprosjekt Byggeplan gjennomført	Thu 01.03.18	Thu 01.03.18		
37	R12 30% av sesjonene I testprosjekt KDP gjennomført	Thu 01.03.18	Thu 01.03.18		
38	R13 60% av sesjonene I testprosjekt Byggeplan gjennomført	Thu 01.11.18	Thu 01.11.18		
39	R14 60% av sesjonene I testprosjekt KDP gjennomført	Thu 01.11.18	Thu 01.11.18		
40	R15 Testprosjekt Bygeplan-SP avsluttet	Mon 01.04.19	Mon 01.04.19		
41	R16 Testprosjekt KDP-SP avsluttet	Wed 01.05.19	Wed 01.05.19		
42	R17 Forsknings- og og effektdokumentasjon fra testprosjekter etablert	Sat 01.06.19	Sat 01.06.19		
43	R18 Metodemanual ver 1 etablert	Sun 30.06.19	Sun 30.06.19		
44	R19 Metodeverk for SP implementert i deltakerbedriftene	Mon 30.09.19	Mon 30.09.19		
45	**AKTUELLE TESTPROSJEKTERS GJELDENDE FRAMDRIFT PR DD.**	**Fri 29.01.16**	**Tue 30.04.19**		
46	IC Kleverud-Sørli (byggeplan). Pilot metodeutvikling	Mon 02.05.16	Fri 01.09.17		
47	IC Brumunddal-Lillehammer (forstudie). Pilot metodeutvikling	Fri 29.01.16	Sat 30.09.17		
48	IC Nykirke-Barkåker (byggeplan). Case testing	Mon 02.10.17	Tue 02.04.19		
49	IC Brumunddal-Lillehammer (KDP). Case testing	Mon 02.10.17	Tue 30.04.19		



Project: SP%20Prosjektplan%20
Date: Thu 17.03.16

Task, Split, Milestone, Summary, Project Summary, Inactive Task, Inactive Milestone, Inactive Summary, Manual Task, Duration-only, Manual Summary Rollup, Manual Summary, Start-only, Finish-only, External Tasks, External Milestone, Deadline, Progress, Manual Progress

Figur 3: Hovedaktiviteter og milepæler

Det refereres til Gantt-diagrammet på forrige side for oversikt over deloppgaver, sammenhenger og tidsplan for hovedaktivitetene. I tabellen nedenfor beskrives hovedaktivitetene på overordnet nivå. Alle de 5 hovedaktivitetene understøtter prosjektets hovedmål og ansees derfor å ligge i FoU-kategorien ***Industriell forskning*** med 100%.

Hoved-aktivitet	**Faglig Innhold**	**Viktige leveranser og resultater som skal oppnås**	**Ansvarlig prosjekt-partner**	**Deltagende prosjektpartnere**	**FoU kategori**	**Tot. kostnad** (1000 NOK)
H1 Prosjekt-ledelse og Styring	Prosjektledelse og styring av forskningsprosjektet, for å sikre at prosjektets resultatmål oppnås, dvs at de definerte leveransene leveres på tid, innenfor budsjett og til riktig kvalitet.	Styringsunderlag, statusrapportering, administrativ sluttrapport inkludert regnskap, for oppnåelse av prosjektleveranser i h.h.t. resultatmål.	Metier	*Metier:* Utarbeide styringsunderlag, oppfølging og rapportering *ViaNova PT:* Input til styringsunderlag og mottaker av rapportering som ansvarlig bedrift	Industriell forskning	539
H2 Forskning	Etablere forskningsmetodikk. Bearbeide eksisterende metode-kunnskap, samt fortløpende analyser av potensiale og oppnådde mål under utviklingen. Evaluere effekter og bidra til implement-ering av de teoretiske resultater i prosessutviklingen.	Teoretisk underlag, målebasis, effektanalyser, dokumentasjon	NTNU	*NTNU:* utarbeide teoretisk underlag, målebasis, effektanalyser, dokumentasjon *VNPT:* Input til teoretisk underlag samt bidra til dokumentasjon. *Epsis:* Bidrag underlag	Industriell forskning	1580
H3 Metode-utvikling	Skal via forskningsbidrag fra aktivitet H2 og teknologisk utvikling fra aktivitet H4 skape en ny metodikk (SP) for samferdselsbransjen. Vil baseres på studier av tidligere prosjekters utfordringer, nye tanker i andre bransjer, samt egen utvikling via pilot- og testprosjekter. SP forutsettes tilpasset og bearbeidet i egen form for alle planfaser i et samferdselsprosjekt. (Utredning, Hovedplan/Kommunedelplan, Detalj-/reguleringsplan og Byggeplan).	Gjennomført pilotprosjekt på to IC-prosjekter Etablert metodehåndbok ver 0 Gjennomført testprosjekt på to IC-prosjekter Etablert metodehåndbok ver 1	ViaNova PT	*ViaNova PT:* Ansvarlig for en pilot og ett testprosjekt. Ansvarlig for metodeutvikling *Rambøll:* Ansvarlig for en pilot og ett testprosjekt. Input til metodeutvikling *VNS:* Ansvarlig for grensesnitt mot verktøy, input til metodeutvikling *NTNU, METIER, Epsis, JBV:* Input til metodeutvikling	Industriell forskning	4704

Hoved-aktivitet	Faglig Innhold	Viktige leveranser og resultater som skal oppnås	Ansvarlig prosjekt-partner	Deltagende prosjektpartnere	FoU kategori	Tot. kostnad (1000 NOK)
H4 Verktøy-utvikling	I tett samspill med behovsidentifikasjon under metodeutviklingen i aktivitet H3 skal det utvikles og implementeres funksjonalitet som understøtter nødvendig funksjonalitet for SP. En viktig premissgiver for dette arbeidet er gjenbruk av eksisterende 3.parts verktøy for å sikre rask implementering og nødvendig grad av fleksibilitet. Utviklingen vil foregå med iterasjoner på ca. 3mnd tilpasset programvarens releaseplan. Etter hver iterasjon blir prioriteringslisten over ønsket funksjonalitet vurdert på nytt basert på praktiske erfaringer fra SP prosessen de siste 3 mnd. Dette sikrer at verktøyutviklingen hele tiden fokuserer på det som er viktigst for brukerne.	Videreutvikling av Epsis TeamBox for å støtte SP for infrastruktur. Tilpasninger og videreutvikling av Novapoint^{DCM} for å støtte alle aspekter ved SP Identifikasjon og evt. tilpasninger av 3. part programvare for å støtte aspekter ved SP som ikke er dekket av løsninger i TeamBox og Novapoint^{DCM}	ViaNova Systems AS	*Vianova Systems AS, Epsis AS:* Programvareutvikling *ViaNova Plan og Trafikk AS, Rambøll AS, NTNU:* Kravspesifikasjon, prioritering, testing	Industriell forskning	12107
H5 Formidling og spredning	Gevinstrealiseringen av forsknings-prosjektet avhenger av at SP tas i bruk i fremtidige samferdselsprosjekter. Det innebærer at prosjektets effektmål, resultatmål og status må presenteres og markedsføres til relevante miljøer og aktører. Prosjektet har en ambisjon om å holde disse miljøene og aktørene oppdatert om potensial, utvikling og resultater på en regelmessig basis slik at man står bedre rustet til å ta i bruk de nye metodene og systemene når de er tilgjengelige.	Spredningsplan, presentasjoner, møter, artikler for å oppnå økt kunnskap og bevissthet hos relevante aktører innen samferdselsbransjen	Metier	*Metier:* Ansvarlig for å opprette og følger opp spredningsplanen *Alle:* Ansvarlige for å gi input til spredningsplanen og gjennomføre sine aksjoner	Industriell forskning	2014

b) Sentrale milepæler for FoU-aktivitetene
Se milepæler i oversikt og plan i kapittel 5 / kapittel 7a.

8. Prosjektpartnere og prosjektorganisering
Firmaene og organisasjonene som deltar i dette prosjektet dekker de fleste av de sentrale rollene i en prosjekteringsprosess innenfor samferdsel; dvs byggherre, prosjekterende, prosjekt/prosjekteringsledere og teknologileverandører. I tillegg er NTNU med som forskningsinstitusjon og i kraft av kunnskapsbase på de teoretiske sider i aktivitetene. Aktørene som er med i prosjektet har alle vist gjennom sitt virke at de er helt i front i bransjen med tanke på innovasjon, kunnskap og evnen til å løse de mest komplekse samferdselsprosjektene. Det er to prosjekterende konsulentselskap med i prosjektet for å sørge for at prosessen utvikles på et bredt nok grunnlag til at det ikke blir en firmaspesifikk prosess. Partnergruppen anses som optimal for dette FoU-prosjektet. Se også de elektronisk innsendte Partneropplysningsark for mer utfyllende informasjon om deltakerne:

P1. ViaNova Plan og Trafikk (VNPT) vil være søkerbedrift og sentral i utviklingen av den nye planleggings- og prosjekteringsprosessen (PPM); Samtidig Prosjektering (SP) samt bidra i teknologiutviklingen.

P2. Rambøll vil i likhet med VNPT være sentral i utviklingen av den nye planleggings- og prosjekteringsprosessen (PPM); Samtidig Prosjektering (SP) samt bidra i teknologiutviklingen.

P3. Metier vil bidra i prosessutvikling og markedsimplementering samt være prosjektleder for FoU-prosjektet.

P4. Epsis vil bidra med metodeinnsikt fra oljebransjen, fasiliteringskompetanse, samt videreutvikling av Epsis TeamBox for bruk i SP arbeidssesjoner.

P5. Vianova Systems (VNS) vil bidra med å utvikle nye applikasjoner/funksjonalitet og grensesnitt i Novapoint for SP.

P6. Norges Teknisk-Naturvitenskapelige Universitet (NTNU) vil bidra med litteratursøk og state-of-the-art analyser, sette opp resultatmålene samt etablere forsvarlige emperi-metoder for å måle resultatene av SP-arbeidet i prosjektet. NTNU vil også bidra med å etablere kontakt mot sine internasjonale partnere.

P7. Jernbaneverket (JBV) vil bidra med å legge til rette for å teste ut og tilpasse SP i sine InterCity-prosjekter og sørge for at beslutningstakere er tilstede i SP-sesjonene, noe som er en forutsetning for at dette skal lykkes.

Vi har ikke tatt med utenlandske aktører som partnere i dette prosjektet. Vi mener at antall partnere i prosjektgruppa er optimalt for å få til et effektivt samarbeide og produksjon av resultater. Men det er samtidig veldig viktig å etablere samarbeide med utenlandske aktører for å studere hva de har oppnådd med sine prosessforbedringsprosjekt.

Partner	Navn på partner	Ansvarlig for Hovedaktivitet:	Deltar også i Hovedaktivitet:
P1	ViaNova Plan og Trafikk	H3	H1, H2, H4, H5
P2	Rambøll		H2, H3, H4, H5
P3	Metier	H1, H5	H2, H3, H4
P4	Epsis		H2, H3, H4, H5

P5	Vianova Systems	H4	H2, H3, H5
P6	NTNU	H2	H3, H4, H5
P7	Jernbaneverket		H3

9. Fordeling av prosjektkostnader (i 1 000 kroner)

a) Fordeling av kostnader på Hovedaktiviteter og FoU-kategori:

Målsettingen for dette prosjektet er å utvikle og teste ut helt nye arbeidsmetoder og programvarekomponenter for planlegging og prosjektering i samferdselsbransjen. Kostnadene i dette prosjektet kommer derfor i sin helhet inn under kategorien «Industriell FoU».

Nr.	*Tittel*	*Kostnads-budsjett (1000 kr)*	*Kostnad Industriell FoU*	*Kostnad Eksperimentell utvikling*
H1	Prosjektledelse og styring	539	539	
H2	Forskning	1580	1580	
H3	Metodeutvikling	4704	4704	
H4	Verktøyutvikling	12107	12107	
H5	Formidling og spredning	2014	2014	
Sum		**20944**	**20944**	

b) Kostnader per utførende partner

Partner	*Personal- og indir.kostnader*	*Utstyr*	*Andre kostnader*	*Totalt*
P1. ViaNova Plan og Trafikk	2340		300[1, 2)]	2640
P2. Rambøll	2340		300[1, 2)]	2640
P3. Metier	900		300[1, 2)]	1100
P4. Epsis	4080		200[1, 2)]	4280
P5. Vianova Systems	9984		300[1, 2)]	10284
P6. NTNU [2)]				0
P7. Jernbaneverket [3)]				0
Sum				**20944**

1) Reisekostnader på 100 kNOK er inkludert for å kunne studere utenlandske miljøer innenfor SP
2) Forskningstimer på NTNU blir finansiert gjennom prosjektet fordelt mellom partnerne slik: VNPT: 200, Rambøl: 200, Metier: 100, Epsis: 100, VNS: 200
3) JBV sine kostnader i prosjektet dekkes i sin helhet gjennom JBV sitt ordinære budsjett og er derfor ikke tatt med her. JBV vil bidra i prosjektet med 1500 timer i året

10. Finansiering per partner

Partner	**Egeninnsats**	**Kontanter**	**Totalt**
P1. ViaNova Plan og Trafikk	1795		1795
P2. Rambøll	1795		1795
P3. Metier	748		748
P4. Epsis	2911		2911
P5. Vianova Systems	6993		6993
P6. NTNU			0
P7. Jernbaneverket			0
Søkt Forskningsrådet			6702
Total finansiering (= totale kostnader)			**20944**

11. Øvrige samarbeidsrelasjoner for FoU-aktivitetene

Det vil etableres kontakt med utenlandske fagmiljøer som en del av de aktivitetene som planlegges. Vi har allerede vært i kontakt med Center For Integrated Facility Engineering (CIFE) på Stanford University. Dette er kanskje det mest sentrale miljøet innenfor dette temaet i verden i dag. Vi ønsker derfor å utvikle et samarbeide med denne organisasjonen gjennom dette prosjektet.

Det foregår en god del spennende i Sverige innenfor bygg i regi av bl.a. Veidekke. BIM Alliance Sweden er også aktive på dette området. De sammenstiller resultater og avholder konferanser om temaet. Det vil være naturlig å ta kontakt med disse miljøene.

Lean-miljøet på Stockholm School of Economics med bl.a. Niklas Modig og Pär Åhlstöm er et spennende miljø i denne sammenhengen. De er sentrale i den grunnleggende «flyt-filosofien» som ligger i bunn for SP-prosessen. Det kan være aktuelt å etablere kontakt med dette miljøet som en del av dette prosjektet.

VNS ble i september 2015 kjøpt opp av det USA-baserte internasjonale Trimble-konsernet. Trimble er i en voldsom utvikling hvor de i hurtig tempo kjøper opp selskaper i hele verden for bl.a. å etablere seg i hele verdikjeden for samferdselsprosjekter fra tidligfase , via prosjektering, bygging til drift og vedlikehold. Gjennom de fire årene dette prosjektet vil vare, vil antakelig Trimble kjøpe opp miljøer som har erfaring med SP og som det kan være aktuelt å etablere et samarbeide med.

DEL 3: Realisering av innovasjonen og utnyttelse av resultater

12. Plan for realisering av innovasjonen

For fremdrift, tidspunkter og hovedaktiviteter i realiseringsplanen henvises det til diagram i kapittel 7a. I det etterfølgende beskrives derfor kun milepælene og hvordan oppnåelse av disse vil sikre realiseringen av innovasjonen og utnyttelse av resultatene.

ID	Milepæler	Dato	Kommentar:
R1	Endelig målformulering og prosjektplan	02.05.16	Konkretiseres i prosjektets styringsdokument. Skal sikre entydige mål for prosjektet samt en tydelig og klar aktivitetsplan
R2	Prosjektstyringsbasis etablert	02.05.16	Konkretiseres i prosjektets styringsdokument. Basis for oppfølging av prosjektet.
R3	Spredningsplan etablert	01.06.16	Egen plan. Skal sikre systematisk spredning, kommunikasjon og utnyttelse av resultater
R4	Forskningsmetodikk etablert	08.08.16	Eget dokument. Definerer hvordan effekter skal identifiseres, måles og dokumenteres
R5	Pilotprosjekt Byggeplan-SP avsluttet	03.10.16	Sikrer at en del av metodens elementer prøves ut i under «kontrollerte forhold» og at verdifull erfaring innhentes
R6	Forsknings- og erfaringsdokumentasjon kartlagt	03.04.17	Sikrer at man har etablert en metodikk og målebasis for å påvise effektene av Samtidig Prosjektering
R7	Metodemanual ver. 0 etablert	01.06.17	Metodebeskrivelse. Benyttes som basis for gjennomføring av testprosjekter
R8	Gjennomføringsplan for test Byggeplan-SP	30.08.17	Sikrer at gjennomføringen av testprosjektene er tilstrekkelig planlagt
R9	Gjennomføringsplan for test KDP-SP	30.08.17	Sikrer at gjennomføringen av testprosjektene er tilstrekkelig planlagt

R10	Pilotprosjekt Forstudie til KDP avsluttet	02.10.17	Sikrer at en del av metodens elementer prøves ut i under «kontrollerte forhold» og at verdifull erfaring innhentes
R11	30% av sesjonene i testprosjekt Byggeplan gjennomført	01.03.18	Sikrer tilstrekkelig fremdrift i gjennomføringen av testprosjektet
R12	30% av sesjonene i testprosjekt KDP gjennomført	01.03.18	Sikrer tilstrekkelig fremdrift i gjennomføringen av testprosjektet
R13	60% av sesjonene i testprosjekt Byggeplan gjennomført	01.11.18	Sikrer tilstrekkelig fremdrift i gjennomføringen av testprosjektet
R14	60% av sesjonene i testprosjekt KDP gjennomført	01.11.18	Sikrer tilstrekkelig fremdrift i gjennomføringen av testprosjektet
R15	Testprosjekt Byggeplan-SP avsluttet	01.04.19	Sikrer ferdigstillelsen av testprosjektet som basis for endelig versjon av metoden.
R16	Testprosjekt KDP-SP avsluttet	01.05.19	Sikrer ferdigstillelsen av testprosjektet som basis for endelig versjon av metoden.
R17	Forsknings- og effektdokumentasjon fra testprosjekter etablert	01.06.19	Sikrer at effektene av samtidig prosjektering dokumenteres
R18	Metodemanual ver. 1 etablert	30.06.19	Beskrivelse av endelig versjon av metoden for Samtidig Prosjektering
R19	Metodeverk for SP implementert i deltakerbedriftene	30.09.19	Sikrer at metoden tas i bruk i Partnerbedriftene

13. Risikoelementer

Risikoprofilen er noe forskjellig for metodeutvikling og verktøyutvikling.

Følgende hovedmomenter er identifisert knyttet til risiko for realisering:

Risikoelement	Strategi for håndtering
Det utvikles andre systemer og prosesser for planlegging, prosjektering og kontrahering som gjør dette mindre relevant, eksempelvis som et resultat av revisjon av lovverket	Utenfor vår kontroll. Risikoen overvåkes
Andre nasjonale og internasjonale aktører som jobber med tilsvarende systemer (som fordrer at en må vurdere timingen i forhold til publisitet i markedet)	Overvåke utviklingen i bransjen. Vurdere timing i forhold til publisering av resultater
Generell endringsmotstand hos aktører i Samferdselsbransjen,	Sentrale aktører identifiseres og det utarbeides en kommunikasjonsstrategi mot hver enkelt) i formidlingsplanen
Begrenset konkurranse blant rådgiverbedriftene og systemleverandørene i en startfase gjør at Byggherrene ikke ønsker å ta den nye metodikken i bruk i stor skala, men holder fast på gammel metode.	Vurdere markedsituasjonen fortløpende. Spre metoden til andre rådgivere dersom markedssituasjonen tilsier det

Manglende vilje til å prøve ut nye metoder hos Byggherrerens organisasjon grunnet økt risiko	Gjennomføre pilottest av metode på «på ikke reelt prosjekt». Tett dialog med JBV sentralt og med prosjektene om metoden
Mangelfull teknologisk kompetanse hos aktører i pilotprosjektet til å kunne jobbe i h.h.t. nye prosesser	Utarbeide metoden slik at den blir uavhengig av prosjektets teknologikompetanse (hos JBV) Rambøll og VNPT er engasjert i prosjektene
Myndighetskrav og offentlige krav knyttet til forvaltning og godkjennelse av planer	Utenfor vår kontroll, men det vil bli gitt innspill til Myndigheter om potensielle forbedringer
Implementering utover landegrensene utfordrende	Ambisjoner og strategi for spredning utover landegrenser fastlegges av hver enkelt partnerbedrift.
Klarer ikke skape nok etterspørsel for metode og verktøy til å få realisert potensiale i prosjektet	Drive aktiv spredning av informasjon om metode og gevinster inn mot nasjonale byggherrer for infrastruktur. Dokumenteres i formidlingsplanen
Manglende vilje til å organisere samferdselsprosjekter i henhold til den nye metodikken	Drive aktiv spredning av informasjon om metode og gevinster inn mot nasjonale byggherrer for infrastruktur. Dokumenteres i formidlingsplanen
Liten eller ingen risiko knyttet til finansiering for selve utviklingsfasen, men dersom kommersialisering i etterkant av prosjektet krever ytterligere investeringer er finansiering uklart	Behov for evt. ytterligere investeringer til kommersialisering avklares i løpet av, og 2017 og strategi legges.

14. Øvrig samfunnsøkonomisk nytteverdi

Den primære samfunnsøkonomiske nytteverdien av forskningsprosjektet er tidligere realisering og ferdigstillelse av samferdselsprosjekter (forutsatt at disse har positiv samfunnsøkonomisk nytteverdi) Dagens situasjon er at de aller fleste samferdselsprosjekter ferdigstilles etter at det reelle behovet som skal dekkes er oppstått.

For eksempel gjør befolkningsveksten på Østlandet at det allerede om får år ikke vil være kapasitet nok til å håndtere alle passasjerene som ønsker å ta toget. IC-utbyggingen skal ta høyde for denne befolkningsveksten, men hele prosjektet er først planlagt ferdigstilt i 2030. Det innebærer at raskere ferdigstillelse vil kunne gi økt kapasitet på et tidligere tidspunkt og dermed tidligere realisering av samfunnsøkonomiske gevinster.

Metodikken skal også bedre kvaliteten på arbeidsprosessen og endelige løsninger, noe som vil gi mindre feil og omarbeiding i byggefasen, og igjen billigere prosjekter.

Øvrig samfunnsøkonomisk nytte vil kunne være:

- At andre sektorer, for eksempel innen bygg, tar metodikken og systemene i bruk og dermed reduserer gjennomføringstid og tidligere kan hente ut nytteverdien.
- Økt interesse for samferdselsfagene på yrkesskoler, høyskoler og universiteter
- Bedre flyt og visualisering av fagutfordringer kan gi bedre forståelse og avklaringstempo i offentlig behandling

15. Formidling og kommunikasjon

Formidling og kommunikasjon er identifisert som en av suksessfaktorene for å lykkes med å realisere gevinsten av prosjektet og det er derfor opprettet en egen arbeidspakke på dette. Prosjektet vil også etablere en kommunikasjonsplan for å sikre strukturert og systematisk formidling og kommunikasjon til alle prosjektets hovedinteressenter. En foreløpig analyse har identifisert følgende hovedinteressenter og tilhørende kommunikasjonskanaler:

Interessent-gruppe	**Spredningskanaler**						
	Vitenskapelige publikasjoner	Samferdsels-konferanser	Møter	Media	Brosjyre-materiell	Internett-side	Hånd-bøker/ metode-beskrivelser
Offentlige Byggherrer		X	X	X	X	X	X
Entreprenører		X		X	X	X	X
Rådgivende ingeniører		X		X	X	X	X
Plan- og bygnings myndigheter			X	X			
Utdannings institusjoner /Akademia	X			X			X
Internasjonale samferdsels-aktører	X	X					

Interessentanalysen og spredningsplanen vil detaljeres nærmere i oppstartsfase av prosjektet.

DEL 4: Øvrige opplysninger

16. Miljøkonsekvenser

Selve prosjektgjennomføringen vil ikke ha miljøkonsekvenser.

Det anses likevel at metoden for samtidig prosjektering i samferdselsprosjekter overordnet sett vil gi positive miljøeffekter siden prosessen vil ha fokus på samhandling og tilstedeværelse av beslutningstagere og interessenter. Det vil øke sannsynligheten for at alle relevante parter kommer til orde og blir hørt på riktig sted i prosessen og at man derigjennom sikrer et riktig beslutningsgrunnlag. Det øker igjen sannsynligheten for gode valg også med tanke på miljø.

17. Etikk

Vi har gjennomgått den refererte forskningsetiske sjekklisten og kan ikke se at prosjektet Samtidig Prosjektering berører etiske forhold i særlig grad. Prosessen vil bidra til større involvering av interessenter og beslutningstagere og gi mer transparens i beslutninger. Dette vil fra prosjektets ståsted kun ha en positiv innvirkning i forhold til etikk.

18. Rekruttering av kvinner, kjønnsbalanse og kjønnsperspektiv

Kjønnsperspektivet ansees ikke som relevant i dette forskningsprosjektet. Det er uvisst hvordan en ny prosess og nye verktøy for Samtidig Prosjektering vil påvirke rekruttering av kvinner til samferdselsprosjekter.

19. Utlysningsspesifikke tilleggsopplysninger

Ingen

20. Fordeling av kostnader på prosjektpartnere

Tabellene nedenfor er utarbeidet sammen med samarbeidspartnerne i prosjektet, og hver enkelt partner (støttemottaker) har godkjent kostnadsfordelingen.

Tabell 1		*Kostnader for oppgavene som hver partner utfører (1000 kr)*										
	Hovedaktivitet	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	Total
Type (kategori) forskning: *Enten* IF *eller* EU		IF	IF	IF	IF	IF						
1 a) Prosjektkostnader hos partnere som er Støttemottakere	Bedrift: Stor, Mellom, Liten: (S/M/L)											
ViaNova Plan og Trafikk AS	Mellom	264	104	1 564	254	254						2 440
Rambøll AS	Stor	-	104	1 828	254	254						2 440
Metier AS	Stor	275	30	320	50	325						1 000
Epsis AS	Liten	-	348	418	2 777	637						4 180
Vianova Systems AS	Stor	-	354	494	8 732	504						10 084
Bedrift 6												-
Bedrift 7												-
Kostnader hos partnere som er støttemottakere		539	940	4 624	12 067	1 974	-	-	-	-	-	20 144
1 b) Prosjektkostnader hos FoU-institusjon eller foretak som leverer FoU-tjenester til prosjektet - og som dermed ikke er støttemottakere												
NTNU		-	640	80	40	40						800
FoU-inst / leverandør 2												-
FoU-inst / leverandør 3												-
FoU-inst / leverandør 4												-
FoU-inst / leverandør 5												-
Kostnader hos partnere som *ikke* er støttemottakere		-	640	80	40	40	-	-	-	-	-	800
Sum prosjektkostnader		539	1 580	4 704	12 107	2 014	-	-	-	-	-	20 944

Tabell 2		*Fordeling av kostnader for oppgaver utført av partnere som ikke er støttemottakere (1000 kr)*										
Støttemottakere	*Hovedaktivitet*	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	Total
ViaNova Plan og Trafikk AS			160	20	10	10						200
Rambøll AS			160	20	10	10						200
Metier AS			80	10	5	5						100
Epsis AS			80	10	5	5						100
Vianova Systems AS			160	20	10	10						200
Bedrift 6												-
Bedrift 7												-
Sum kostnader som må dekkes av støttemottakerne		-	640	80	40	40	-	-	-	-	-	800
Kontrollsum - de gule radene skal være like		✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -

Tabell 3: Fordeling av prosjektkostnader per støttemottaker og Hovedaktivitet. Tall summeres automatisk fra Tabell 1a) og Tabell 2.

		Prosjektkostnader for hver støttemottaker (1000 kr)										
Støttemottakere	*Hovedaktivitet*	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	Total
	Type FoU	IF	IF	IF	IF	IF	0	0	0	0	0	
ViaNova Plan og Trafikk AS	Egne kostnader	264	104	1 564	254	254	-	-	-	-	-	2 440
	Kostnader andre	-	160	20	10	10	-	-	-	-	-	200
	Sum for bedrift	*264*	*264*	*1 584*	*264*	*264*	-	-	-	-	-	*2 640*
Rambøll AS	Egne kostnader	-	104	1 828	254	254	-	-	-	-	-	2 440
	Kostnader andre	-	160	20	10	10	-	-	-	-	-	200
	Sum for bedrift	-	264	1 848	264	264	-	-	-	-	-	2 640
Metier AS	Egne kostnader	275	30	320	50	325	-	-	-	-	-	1 000
	Kostnader andre	-	80	10	5	5	-	-	-	-	-	100
	Sum for bedrift	275	110	330	55	330	-	-	-	-	-	1 100
Epsis AS	Egne kostnader	-	348	418	2 777	637	-	-	-	-	-	4 180
	Kostnader andre	-	80	10	5	5	-	-	-	-	-	100
	Sum for bedrift	-	428	428	2 782	642	-	-	-	-	-	4 280
Vianova Systems AS	Egne kostnader	-	354	494	8 732	504	-	-	-	-	-	10 084
	Kostnader andre	-	160	20	10	10	-	-	-	-	-	200
	Sum for bedrift	-	514	514	8 742	514	-	-	-	-	-	10 284
Bedrift 6	Egne kostnader	-	-	-	-	-	-	-	-	-	-	-
	Kostnader andre	-	-	-	-	-	-	-	-	-	-	-
	Sum for bedrift	-	-	-	-	-	-	-	-	-	-	-
Bedrift 7	Egne kostnader	-	-	-	-	-	-	-	-	-	-	-
	Kostnader andre	-	-	-	-	-	-	-	-	-	-	-
	Sum for bedrift	-	-	-	-	-	-	-	-	-	-	-
Sum prosjektkostnader		539	1 580	4 704	12 107	2 014	-	-	-	-	-	20 944
Kontrollsum - grønne rader skal være like		✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -	✔ -

21. Opplysninger om annen offentlig støtte

Det er ikke tildelt annen offentlig støtte til prosjektet. Samtlige bedriftspartnere har bekreftet at de ikke er blitt tildelt offentlig støtte for kostnader de vil ha til gjennomføring prosjektet.

Referanser

Andersen, Bjørn, Belay, Alemu Moges og Amdahl Seim, Eva: "Lean Construction Practices and its Effects: A Case Study at St Olav's Integrated Hospital, Norway", Lean Construction Journal, Issue 2012, pp. 122-149

Ballard, Glenn (2000). "Lean Project Delivery System." White Paper #8, Lean Construction Institute, May 1.

Ballard, Glenn (2008)."The Lean Project Delivery System: An Update", Lean Construction Journal, pp. 1-19.

Koskela, L. (1992). "Application of the New Production Philosophy to Construction". Technical Report # 72, Center for Integrated Facility Engineering, Department of Civil Engineering, Stanford University, CA.

Love, Peter E.D., Lopeza, Robert, Edwards, David J., Goh, Yang M. (2012). "Error begat error: Design error analysis and prevention in social infrastructure projects", Accident Analysis and Prevention, Vol. 48, pp. 100–110.

Love, Peter E.D., Smith, Jim, Simpson, Ian, Regan, Michael, Olatunji, Oluwole. (2014) " Understanding the landscape of overruns in transport infrastructure projects", Environment and Planning B: Planning and Design 2014, volume 41.

Smith, Robert P.. "The Historical Roots of Concurrent Engineering Fundamentals". IEEE Transactions On Engineering Management, Vol. 44, No. 1, February 1997.

Tribelsky, Effi, Sacks, Rafael. (2011) "An Empirical Study of Information Flows in Multidisciplinary Civil Engineering Design Teams using Lean Measures", Architectural Engineering and Design Management, Vol. 7, Iss. 2, pp. 85-101-

Welde, Morten, Samset, Knut, Andersen, Bjørn og Austeng, Kjell: Lav prising – store valg: En studie av underestimering av kostnader i prosjekters tidligfase, Concept rapport nr. 39, 2014

Wodalski, Michael J., Thompson, Benjamin P., Whited, Gary, Hanna, Awad S. (2011). "Applying Lean Techniques in the Delivery of Transportation Infrastructure Construction Projects", National Center for Freight & Infrastructure Research & Education Department of Civil and Environmental Engineering, College of Engineering, University of Wisconsin–Madison.

Appendix D

Notes from Jira Workshop with Hæhre Betonmast

2018.04.23 Møte med Trimble og Cowi

mandag 23. april 2018 07:21

Møte

Issue typer (JIRA har et valg som kan skru av send til EA)
- Møtereferater
  - Kjartan sender over møtereferat til BetonmastHæhre som Trimble legger til grunne.
    - Trimble gir innspill.
- Konflikter/Modell kommentarer (EA)
- Beslutninger Interne (EA)
- Beslutninger Eksterne (NV + Kommune++) (EA)
- HMS / YM (EA)
- KS (EA)

Prioritet
- Ikke satt ------- (blir automatisk satt til dette)
- Normal
- Høy (Kritisk for fremdrift/økonomi)

- Beslutningsnivå
  - Rådgiver
  - Entreprenør
  - Bygghere

- PNS (Epic)
  - Kjartan sender over. (12 Parseller)

- Fag (Component)
  - Veg
  - VA
  - …
  - Adrian sender over.

- Milepels Leveranser (FIX version)
  - Kjartan sender over..

- Sprint
  - 7 dagers intervall
    - 2018 - Uke 01
    - ….
    - 2018 - Uke 17
    - 2018 - Uke 18
    - 2018 - Uke 19
    - 2018 - Uke 20

- Medlemmer
  - Kjartan sender over liste med deltagere for gjennomføring

- Do Date

- Eget punkt (Issue type interne og eksterne beslutninger. Kommer det fra konflikt i modell må man lage en ny sak i JIRA som har da har beslutningsflyt opp mot NV, etc. .)
  - Endringsorder
  - Fravik fra kontrakt (KS)
  - Fravik fra Normalene

- Estimat på oppgaver (skal ikke brukes aktivt i dette prosjektet men testes ut)
  - Konsulent setter estimert timer på oppgaver
  - Brukte timer
  - Gjenstående timer
  - Kobling mot Maconomy (senere)

- Mail svar til JIRA fra mailklient. (ønsker om dette etter hvert)

Gemini ser på dette Trimble har vært i kontakt Hæhre må ta kontakt og følge op.
- Trimble må se på hvordan de transformere til lokalt null.

BCF2 import og eksport i løsningen.

Kanban.
- Møtereferater (Ser på Sweco malen og Hæhre/Cowi gir innspill)
  - Planlagte / Nytt møte
  - Møtereferat

    - Godkjent møtereferat

- Konflikter/Modell kommentarer (EA)
    - Skal utføres (To Do)
    - Under arbeid
    - Klar til modell gjennomgang
    - Godkjent (Bare BIM, PL og Hæhre som kan sette)

- Beslutninger Interne (EA),   Beslutninger Eksterne (NV + Kommune++) (EA)
    - Skal utføres (To Do)
    - Under arbeid
    - Sendt til beslutningstaker
    - Besluttet og lukket

- HMS / YM (EA)
    - Skal utføres (To Do)
    - Under arbeid
    - Tiltak

- KS (EA)
    - Skal utføres (To Do)
    - Under arbeid
    - Godkjent

Diagrammer:

Tar opp eget møte på dette.

Aksjoner

Cowi rydder EA

Hæhre + Cowi sender over saker over i løpet av 23/04-2018

Trimble konfigurer ny JIRA databasen (sammen med Cowi)

- Denne skal være mulig å flytte til BetonmastHæhre.

BetonmastHæhre og Cowi jobber med å få ett forslag til løsning 27/4-2018

Kjartan og Adrian sender brukere som skal ha tilgang til JIRA til Øystein

Steinar og Anders sender mail til Gemini og etterspør kobling til JIRA.

Anders bakgrunn til å bruke Sharepoint og Solibri samt tankene her.

Steinar sjekker ut server hos Hæhre (husk mail klient)

Appendix E

Description of SPP by Jan Erik Hoel

Trimble

Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Bakgrunn for prosjektet

Jan Erik Hoel <[redacted]> Tue, May 22, 2018 at 7:39 AM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>
Cc: Erling Graarud <[redacted]>

Hei!

Se tekster under.

Igjen skriver jeg på engelsk så det blir lettere å bruke mer direkte hvis du ønsker det.

Jan Erik

From: Oystein Bjerke Hauan [mailto:oysteinbjerke_hauan@trimble.com]
Sent: 7. mai 2018 13:54
To: Jan Erik Hoel <[redacted]>
Cc: Erling Graarud <[redacted]>
Subject: Bakgrunn for prosjektet

Hei,

det hadde vært til stor hjelp for oppgaven min om du/dere kunne skrevet en liten tekst/introduksjon til SPP prosjektet og IssueConnector.

- Hvorfor er det et behov for å endre planleggings og prosjekteringsfasen?

JEH:

There is and has been a requirement from the last governments (2xStoltenberg and 2xSolberg) to reduce the planning, design and build time for infrastructure project with 50%. Up to now the average timespan from planning to finished infrastructure system has been about 10 years.

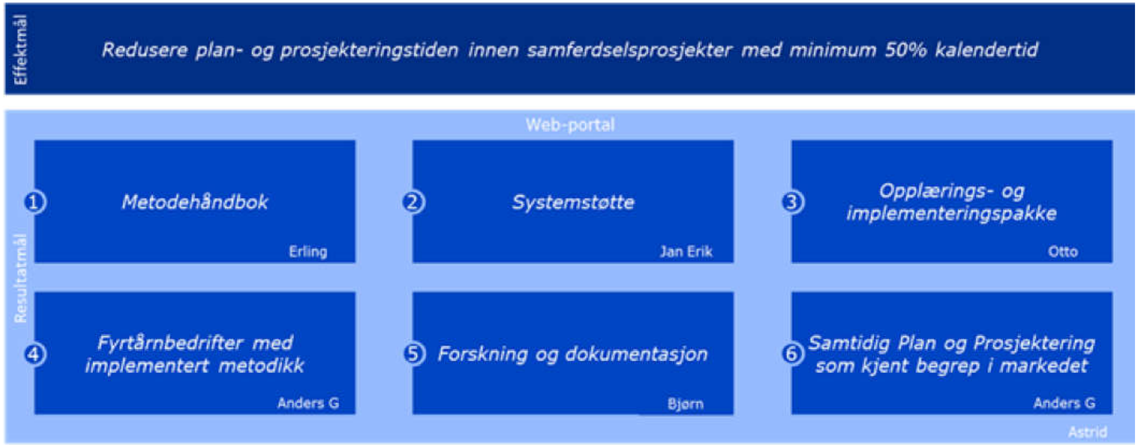
There are initiatives to re-organize the official planning-process (“plan og bygningsloven”), but at the same time there are several initiatives to improve the design and construction process itself.

Maybe the most important initiative has been to establish Nye Veier (NV). NV represents a total shift in the industry regarding contract regimes and requirements to the contractors and engineers. Early involvement of the contractor, larger contracts, trust based contracts, focus on BIM-model, requirements to NOT use drawings are among the requirements that really push the industry in a new direction. The other public infrastructure organizations like Statens vegvesen and Bane NOR also follow this new way of thinking project organization.

- Helt konkret: hva gjør SPP-"gruppen" for å forbedre dette?
JEH:

- Collect knowledge and best practises of how to organize concurrent sessions and other processes that support the concurrent sessions (stake holder analysis, risk analyses etc)
- Develop a digital handbook (web pages) for best practises
- Develop software solutions to support the processes
- Test out the processes in real projects
- Do research to establish base line and measure the improvements
- Inform the industry about the findings through workshops, seminars, articles etc

- Here is the overview that Otto has made:



- Hvordan kan IssueConnector og JIRA være med på å løse noen av problemene, og i så fall hvilke problemer blir (forhåpentligvis) løst?

JEH – se også forrige mail jeg sendte om bakgrunnen for IC

Due to the shift in the industry to a more agile and fast integrated design and construct process, new decision making and design conflict handling processes are emerging. In larger projects there are several thousand issues to handle. The tools to handle these issues must be fast, agile and intuitive. The best agile tools for process control / case handling are developed for the software industry. There are many of these tools on the marked today with pro and cons – and many more will be launched in the years to come.

An easy to configure loose coupling between different process control systems and different comment/view point systems for BIM models is a key component to solve the emerging agile work process in integrated teams between project owners, contractors and engineers.

- Hvordan ser Trimble for seg å bruke utfallet av denne oppgaven?

JEH: Trimble will develop IC further to make robust product. We still have to decide if the solution will be sold as a standalone product or if it will an integrated component in our Novapoint/Quadri solution.

We have set up a meeting series now in the spring of 2018 to decide the further productification process for IC. We will probably deploy the solution on a cloud service.

Skriv gjerne noen tanker rundt dette, det trenger ikke være sammenhengende nødvendigvis. Dette skal jeg bruke for å formulere problemstillingen og forsvare viktigheten av den, samt beskrive utfordringen i samferdselssektoren idag.

Mvh

Øystein Hauan

Appendix F

Evaluation of Jira and IssueConnector from Hæhre Betonmast and Cowi

Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Evaluering av IssueConnectors potensiale

6 messages

Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com> Mon, May 21, 2018 at 1:34 PM
To: Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>, "Steinar G. Rasmussen" <Steinar.Rasmussen@akh.no>, Adrian Saunders <adsd@cowi.com>, Ulf Kristiansen <ulf.kristiansen@ramboll.no>, Kjartan Kristoffersen <kjartan.kristoffersen@akh.no>

Hei!

Jeg er nå i siste innspurt på oppgaven min, og tenkte det kunne vært fint med litt tilbakemeldinger på IssueConnector fra dere som har prøvd det litt.

Så jeg hadde blitt veldig glad hvis dere kunne svart på følgende spørsmål:

1) Hvilket potensiale tror du en løsning som IssueConnector har i infrastruktur-bransjen?
Hvis man ser for seg at man kan koble på flere ulike systemer, som gemini connected, BimCollab osv?

2) Hva er bra med IssueConnector per idag?

3) Er det noe som bør forbedres før den kan tas i bruk på ordentlig?

Oppgaven skal inn om 10 dager så hvis dere har tid til å svare i løpet av idag eller imorgen hadde det vært supert!

Mvh
Øystein Hauan

Steinar G. Rasmussen <Steinar.Rasmussen@akh.no> Mon, May 21, 2018 at 2:53 PM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Gir noen innspill.

1) Hvilket potensiale tror du en løsning som IssueConnector har i infrastruktur-bransjen?

Hvis man ser for seg at man kan koble på flere ulike systemer, som gemini connected, BimCollab osv?

Dette kan være programmet som prosjekt blir styrt igjennom. (JIRA)

Saker i Novapoint er en liten del av et stort prosjekt. Får du koblet til f.eks. Gemini, navisworks, tekla, mm vil det være en hjelp, men den store gevinsten får man først når man styrer hele prosjektet igjennom denne løsningen. Så jeg vil si at dette er en av de lavthengende fruktene for å få innført å tatt i bruk systemet. Men man vil virkelig få den store effekten når man kjører prosjektet gjennom JIRA.

Så tror dette er med på å legge til rette for en ny arbeidsmetode, det å koble på flere programmer er bare første steget. Den virkelig endringen får du når du kjører og styrer all kommunikasjon igjennom JIRA.

2) Hva er bra med IssueConnector per idag?

Det er 1:1 mellom JIRA og NP og begge to blir oppdatert samtidig du har hele tiden den samme informasjon.

Det viser at konseptet fungerer og at vi må få flere til å koble seg til.

Og at vi får startet å bruke programmet.

3) Er det noe som bør forbedres før den kan tas i bruk på ordentlig?

Dette er en endringsprosess vi må jobbe med opplegget inne i JIRA for å få det lett forståelig.

Men sånn som løsningen foreligger i dag er det fult mulig å starte å bruke den. Men utfordringen som jeg ser den er opplæring og det å få brukerne i prosjektet i gang med å bruke programmet.

Med vennlig hilsen

BETONMASTHÆHRE ANLEGG AS

Steinar G. Rasmussen

Prosjektutvikler / BIM ansvarlig Divisjon Anlegg

Mobil: +47 412 12 488

Steinar.rasmussen@akh.no I bmhe.no

From: Oystein Bjerke Hauan [mailto:oysteinbjerke_hauan@trimble.com]
Sent: mandag 21. mai 2018 13:34
To: Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>; Steinar G. Rasmussen <Steinar.Rasmussen@akh.no>; Adrian Saunders <adsd@cowi.com>; Ulf Kristiansen <ulf.kristiansen@ramboll.no>; Kjartan Kristoffersen <kjartan.kristoffersen@akh.no>
Subject: Evaluering av IssueConnectors potensiale

[Quoted text hidden]

Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com> Mon, May 21, 2018 at 3:25 PM
To: "Steinar G. Rasmussen" <Steinar.Rasmussen@akh.no>

Supert!

Takk skal du ha:)

Mvh
Øystein Hauan
[Quoted text hidden]

Jan Erik Hoel <jan_erik_hoel@trimble.com> Tue, May 22, 2018 at 6:51 AM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Hei!

Se svar under.

Jeg svarer på engelsk, så kan du evt. bruke dette mer direkte i oppgaven din.

Jan Erik

From: Oystein Bjerke Hauan [mailto:oysteinbjerke_hauan@trimble.com]
Sent: 21. mai 2018 13:34
To: Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>; Steinar G. Rasmussen <Steinar.Rasmussen@akh.no>; Adrian Saunders <adsd@cowi.com>; Ulf Kristiansen <ulf.kristiansen@ramboll.no>; Kjartan Kristoffersen <kjartan.kristoffersen@akh.no>
Subject: Evaluering av IssueConnectors potensiale

Hei!

Jeg er nå i siste innspurt på oppgaven min, og tenkte det kunne vært fint med litt tilbakemeldinger på IssueConnector fra dere som har prøvd det litt.

Så jeg hadde blitt veldig glad hvis dere kunne svart på følgende spørsmål:

1) Hvilket potensiale tror du en løsning som IssueConnector har i infrastruktur-bransjen?

Hvis man ser for seg at man kan koble på flere ulike systemer, som gemini connected, BimCollab osv?

JEH: The way of organizing the design work for infrastructure projects in Norway is changing. It follows the same change path as software development processes did 20 years ago when agile methods were introduced. The design processes are shifting from so-called "waterfall processes" (design – build – verification) to agile methods with many minor iterations. Design and build is more in parallel than before. Today the contractor start to build the infrastructure systems before the detailed design is done by the engineers.

Due to this process shift, the industry needs new effective, agile ways of communicate all the issues and decisions that arise during the design and build process. This new design process is very similar to agile software development processes and therefore some of the innovative companies in the business are trying to use "case handling systems" from the software industry.

From before the companies are using "comments" in BIM-models as a rudimentary agile case handling system. To connect the existing "comment" systems in different BIM-products with agile process tools is an obvious requirement from many of these companies.

There are many agile process tools on the market and there are many different BIM- and design tools in use for infrastructure development. A loose coupling between the different systems that can be set up to serve the specific needs for every project is the best way of solving this. A solution like Issue Connector has a great potential to be a very useful "backbone" in the agile design process in the future.

2) Hva er bra med IssueConnector per idag?

JEH:

- It fulfil most of the highest ranked requirements from the customers.
- It is flexible (easy to connect to new systems)
- It is easy to develop further because of a well-designed architecture
- The code base is small (not much extra fancy stuff – just what we need to fulfil the purpose of the system) and well documented
- It is reasonable robust and seems to have good capacity

3) Er det noe som bør forbedres før den kan tas i bruk på ordentlig?

JEH:

- Improve the architecture for handling of many instances of the same product (like Jira e.g.) connected to the same project workflow
- Deploy the solution to the a cloud system like Azure e.g.
- Improve the home page, so that super-users in the different projects can set up and administrate the connections themselves.
- Better handling of users, to avoid as much manual punching of users as possible.

Oppgaven skal inn om 10 dager så hvis dere har tid til å svare i løpet av idag eller imorgen hadde det vært supert!

Mvh

Øystein Hauan

Adrian Saunders <adsd@cowi.com> Tue, May 22, 2018 at 2:35 PM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Hei

1) Vi har veldig troa på at en løsning som dette kan hjelpe oss mye siden vi bruker så mange forskjellige løsninger for å styre et prosjekt. Ting som aksjonslister har tradisjonelt bare vært et Excel-ark, noe som føles ganske primitivt når måten vi jobber på har utviklet seg såpass mye bare de siste tre-fem åra, da det meste var 2D-/tegningsbasert. Med løsninger som dette kan vi koble ting som vanligvis ville vært tekst i en rute i Excel til en "geografisk" plassering i den samme modellen som ingeniøren som skal løses problemet allerede jobber. Her kan vi også ha mer kontroll på hvem som skal gjøre hva og når, og hvem som har tatt avgjørelsen, versjonshistorikker o.l., som det vanligvis ikke har vært så nøye med. Det settes strengere og strengere krav til oss konsulenter (nå om dagen mye pga Nye Veier) om at vi skal dokumentere at vi har kontroll i prosjektene våre når det kommer til økonomi og fremdrift, der vi tradisjonelt bare har sagt at vi har kommet halvveis i prosjekteringen når vi har brukt opp halvparten av pengene. Med løsninger som IssueConnector kan vi (kanskje ikke nå, men etter hvert i teorien) koble sammen oppgaver som må løses i modellen til timelistene våre, for å forbedre den økonomiske styringen i prosjektene. Det hjelper også på kommunikasjonen å tvers at de forskjellige fagområdene i et prosjekt, der f.eks veg/VA bruker Novapoint, prosjektledelsen hovedsakelig bruker Epost/web-baserte løsninger og entreprenøren kanskje foretrekker Gemini/Navisworks, etc.

2) Svarte kanskje litt på denne på forrige spørsmål, men det er vel kanskje hovedsakelig at vi kan knytte forskjellige fagområder som foretrekker forskjellig arbeidsmetodikk sammen i én felles løsning, selv om man ikke bruker samme programvare. På E18 Rugtvedt-Dørdal var det i perioder litt utfordringer med at fagansvarlige (som ikke prosjekterer) ikke så utfordringene det ble meldt fra om i modellen, selv om det faktisk var de som skulle ta avgjørelsen på hvordan vi skulle løse problemet. Easy Access viste seg å ikke være helt ideelt alene, men med IssueConnector kan vi også benytte oss av strykene til allerede eksisterende løsninger (i dette tilfellet Jira) i stedet for å vente på at Trimble skulle utvikle sin helt egen løsning, som ville blitt litt langt ut på siden i forhold til hva Novapoint faktisk er, og som igjen kanskje ikke ville snakket med programvaren til Autodesk eller Powel.

3) Stabiliteten på løsningen kunne vel kanskje vært bedre, men dette er vel kanskje ikke selve IssueConnectoren sin feil. Vi har kanskje ikke fått testet dette nok til å oppdage problemene til selve IssueConnectoren, vi merker mer til begrensningene i Easy Access/Jira. Vet ikke hvor realistisk det er, men jeg tror også at løsningen må være såpass enkel å bruke at vi kan sette opp nye Jiraprosjekter (f.eks) og koble disse opp mot Quadri/Easy Access selv uten hjelp av en utvikler. Vi har et nytt prosjekt på trappene som tilsvarer E18-prosjektet, og skulle vi ha gått for en løsning på sikt i COWI med IssueConnector mellom Novapoint og Jira ville vi nok av praktiske/økonomiske årsaker vært avhengige av at disse koblingene kunne settes opp av vårt eget BIM-personell, som gjerne er ingeniører (vanligvis med litt ekstra kompetanse innen data) og ikke utviklere.

Med vennlig hilsen

Adrian Saunders

Ingeniør

Vegplanlegging

COWI

Kobberslagerstredet 2, 1671 Kråkerøy

Postboks 123

1601 Fredrikstad

Telefon: +47 02 69 4# ##

Mobil: +47 928 07 076

Email: adsd@cowi.no

Website: www.cowi.no / www.cowi.com

LinkedIn Facebook Twitter

From: Oystein Bjerke Hauan [mailto:oysteinbjerke_hauan@trimble.com]
Sent: Monday, May 21, 2018 1:34 PM
To: Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>; Steinar G. Rasmussen <Steinar.Rasmussen@akh.no>; Adrian Saunders <adsd@cowi.com>; Ulf Kristiansen <ulf.kristiansen@ramboll.no>; Kjartan Kristoffersen

<kjartan.kristoffersen@akh.no>
Subject: Evaluering av IssueConnectors potensiale

Hei!

[Quoted text hidden]

Kjartan Kristoffersen <kjartan.kristoffersen@akh.no> Tue, May 22, 2018 at 5:58 PM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

1) Hvilket potensiale tror du en løsning som IssueConnector har i infrastruktur-bransjen?

Hvis man ser for seg at man kan koble på flere ulike systemer, som gemini connected, BimCollab osv?

- Potensialet er en kommunikasjonsplattform for alle prosjektets faser og prosjektdeltakeres nivå.

2) Hva er bra med IssueConnector per idag?

- Det er bra at den reagerer så fort og relativt sømløst.

3) Er det noe som bør forbedres før den kan tas i bruk på ordentlig?

- Brukergrensesnittet på JIRA bør bli mer intuitivt. Må være slik at de som ikke driver med direkte prosjektering og ikke er spesielt opptatt av ny teknologi ser på dette som en forenkling. Varslinger og evt. automatisk logging av eposter kan føre til at flere tar dette i bruk uten for store terskler.

Med vennlig hilsen

BETONMASTHÆHRE ANLEGG AS

Kjartan Kristoffersen

Prosjekteringsleder E18 Rugtvedt-Dørdal

Mobil: +47 952 34 176

kjartan.kristoffersen@akh.no | bmhe.no

Fra: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>
Sendt: mandag 21. mai 2018 13.34
Til: Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>; Steinar G. Rasmussen <Steinar.Rasmussen@akh.no>; Adrian Saunders <adsd@cowi.com>; Ulf Kristiansen <ulf.kristiansen@ramboll.no>; Kjartan Kristoffersen <kjartan.kristoffersen@akh.no>
Emne: Evaluering av IssueConnectors potensiale

[Quoted text hidden]

Appendix G

Evaluation of IssueConnector by Rambøll

Ulf Kristiansen <ulf.kristiansen@ramboll.no> Thu, May 24, 2018 at 1:10 PM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>
Cc: Goran Huseinovic <goran.huseinovic@ramboll.no>, Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>

1. Potensiale: Fange spørsmål, problem og løsninger direkte i utviklingsverktøy sammen med god håndtering og prioritering i prosjektstyringsverktøy. Det viktige er å kunne se «det samme bildet» fra begge sider.
2. BimCollab: Er ikke kjent med BimCollab i samferdsels-prosjekter.
3. Per i dag: Selv om vi kun har det en vei er det utrolig fint å slippe å opprette en JIRA-task manuelt og få oppdateringene fra Easy Acces inn i JIRA for håndtering der.
4. Forberedes: Vet ikke om det er tenkt på allerede, men lurer på hvordan sletting av topic skal fungere sammen med «done» i JIRA.

Med vennlig hilsen

Ulf Kristiansen

BIM Samferdsel

M +47 90879701

ulf.kristiansen@ramboll.no

____

Connect with us

Rambøll

Hoffsveien 4
Postboks 427 Skøyen
0213 Oslo

www.ramboll.no

From: Oystein Bjerke Hauan [mailto:oysteinbjerke_hauan@trimble.com]
Sent: 24. mai 2018 11:33
To: Ulf Kristiansen <ulf.kristiansen@ramboll.no>
Cc: Goran Huseinovic <goran.huseinovic@ramboll.no>; Jan Erik Hoel <Jan_Erik_Hoel@trimble.com>
Subject: Re: Nye prosjekter og linke opp topic i etterkant ++

Hei,

jeg tipper det er et sertifikatproblem.

Dere kjører på HTTPS og vi kjører på HTTP, som sikkert gjør at meldingene fra deres Jira ikke kommer frem grunnet manglende sertifikat her.

Trimble må etterhvert få IssueConnector over på HTTPS, og da vil det forhåpentligvis fungere!

Skal se på det når jeg har levert oppgaven.

image004.jpg
1K

Goran Huseinovic <goran.huseinovic@ramboll.no> Thu, May 24, 2018 at 1:12 PM
To: Oystein Bjerke Hauan <oysteinbjerke_hauan@trimble.com>

Hei Øystein,

Vi har så vidt prøvd IssueConnector, men for være helt ærlig så har vi allerede sett mange fordeler, og potensialet er stort! Så stort, at jeg ønsker også å teste dette i et annet prosjekt, der vi faktisk kommer til å involvere Bane Nor sin prosjekteringsleder inn i våre Samhandlingsmøter/prosjekteringsmøter.

Dermed blir mine svar som følge:

1.Som nevnt over, så er potensialet stort. Etter min mening så ligger potensialet i det at man faktisk integrerer samarbeidsmodellen opp mot prosjektet samarbeidsplattform, i f. eks prosjekteringsmøter. Som regel så har man forskjellige lister (excel), der aksjonene noteres å følges opp. Disse lever eget liv ved siden av det som foregår i modell. Dvs. tungvint å følge opp. og sårbart i tilfelle man ikke kan stille i alle prosjekteringsmøter. Bruker du f.eks JIRA som et samarbeidsplattform, så kan topic opprettes av hvem som helst, og følges opp av alle i JIRA.

Dessuten så gjør det mulig med en mer dynamisk prosjektering, der man f. eks kan få umiddelbar svar på aksjonen, uten at den ansvarlige er med i møtet.(forutsetter bruk av JIRA)

Den tredje fordelen er den visuelle biten, som blir mer tilgjengelig for alle, og ikke bare de som bruker modell.

Uansett, så ligger potensialet i det å koble det opp mot prosjektenes samarbeidsplattform. Da snakker vi også om mulighet til å involvere Kunder/Samarbeidspartner som til enhver tid har oversikt over utfordringene.

2. Umiddelbar visualisering av problemet som må rettes opp, tilgjengelig for alle.

To-veis kommunikasjon mellom systemer.

Rask dataoverføring.

3.Må nok testes litt mer, men uansett så er det snakk om minimale endringer for å ta dette i bruk. For våre behov, så kunne jeg godt tatt det i bruk med det samme.

Håper du har fått noen lunde utfyllende svar, som du kan bruke videre Øystein!

Ønsker deg lykke til med oppgaveinnleveringen, og håper nok at vi opprettholder kontakten ifm. videre bruk av IssueConnector😊

Med vennlig hilsen

Goran Huseinovic

Appendix H

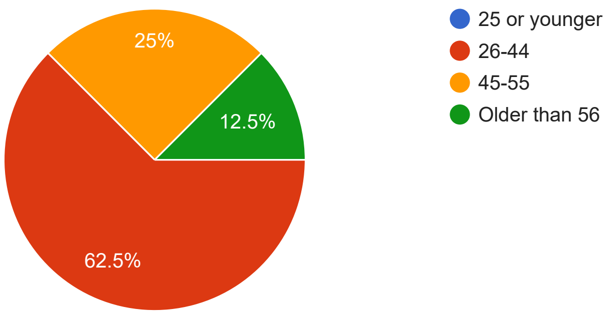
Results from Sweco and Rambøll on Jira Survey

Using JIRA in infrastructure projects - feedback

8 responses

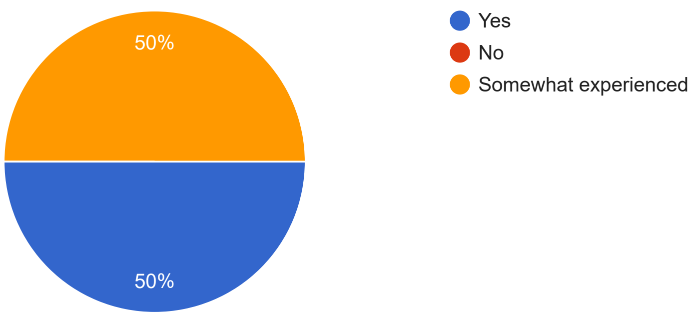
What is your age?

8 responses



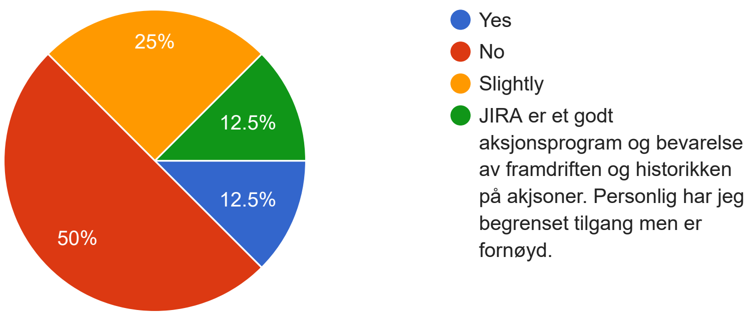
Do you consider yourself an experienced software/computer user?

8 responses



Were you overwhelmed by the user interface of JIRA?

8 responses



Please describe what your field of work is. (Eg. Project manager, BIM-coordinator etc)

8 responses

BIM Coordinator

Project controll

Fagansvarlig

PL-støtte (PL= prosj.ledelse

Project manager / fagansvarlig

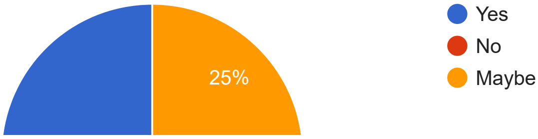
Expert

Fagansvarlig Visualisering

Coordinator

Do you believe JIRA has a potential in the infrastructure industry?

8 responses



Please describe in a few words or sentences your experience with JIRA. Was it positive or negative?

8 responses

I have used JIRA in previous work (computer games industry) and was happy to see that it was also used at my new work place in the infrastructure industry.

Flexible and intuitive. But takes some time to understand the principles of Scrum

Godt aksjonsprogram. Oversiktlig fordeling av arbeidsoppgaver, framdrift og beslutninger. Noe vanskelig brukermåte.

positiv, enkel intuitiv, to the point

Stort sett positiv

i did not know what I was allowed to do

First impressions were a little overwhelming. A lot of text and little visual aids. After getting used to it I have appreciated to have all of the resources and tasks in one place, with the possibilities of commenting and sorting tasks in sprints. Yet I think it could benefit of a better user interface with less clutter/information and more visual aids.

Positive

What are, in your opinion, the main benefits of utilising such software in regards to your field of work?

8 responses

Get a detailed overview over communication, decisions and priorites in project.

Giving all project members the ability to plan short term tasks. All project members are also given insights to the whole project, and to the challenges and workloads of co-workers

Man kan fordeling arbeidsoppgaver på spesifikke tema og/eller fag. Man kan sette opp en framdrift for aksjoner og beslutning. Og man ivaretar historikken for aksjoner gjennom prosjektet.

ha styring med oppgavene, kommunikasjon i prosjekttet

Mulighet for å sette opp små og store konkrete arbeidsoppgaver med klare definerte tidsrammer, adressert til enkelt personer. Oversiktlig måte for å kunne følge progresjonen for hver enkelt oppgave, herunder legge inn oppfølgende kommentarer.

a common place for decisions and outcomes

Having a common platform with all avaiable resources and tasks in one place.

Visual progress and issues for everyone in the project group.

What are, in your opinion, the main drawbacks of utilising such software in regards to your field of work?

8 responses

Unfamilar way of working to many people in the infrastructure industry so it will take time to switch, get it right and make people confidendt.

Challenging to implement. New type of working strategies that may seem overwhelming to some.

Man bør være noe selektiv i valg av aksjoner slik at man ikke lager for mange små og ubetydelig aksjoner.

kan bli mange småting, hoper seg opp, utfordring å velge rett detaljnivå

Kan bli veldig mange oppgaver å forholde seg til for enkelte. En må være bevisst hvem som er adressat. Og mange oppgaver kan overlappe hverandre

you cannot link directly to relevant documents

Jira might feel a little too rigid and unnecessary complicated, especially in my field of work which is a creative discipline (Visualisations).

Some improvements in reliance to some functions so that is more user friendly and easier to use

Do you have any other comments or thoughts on the future potential of JIRA in this industry?

5 responses

Lots of potential to be used as a universal project platform.

Brukervennlighet kan kanskje forbedres.

stre muligheter hvis vi bruke fornuftig

For tidlig å si. Ikke brukt det nok enda.

No

Appendix I

Screenshot of IssueConnector Configuration Page

Please note that the usernames in jira have been striked out. The ID's of each project have also been modified.

Configuration Page

Project Settings

Current project mappings:

JIRA	JIRA Url	Connect Url	Connect	EasyAccess	EasyAccessUrl
CE	http://ic.novapoint.com:8080/	gQQ1GZf00to		636530753371849986	
E18RD	http://ic.novapoint.com:8080/			636561988418328110	
NTL	http://ic.novapoint.com:8080/			636564465655601877	
E18RDP	http://80.232.45.109:8090/			636318189641411636	
NIT	https://projectplanner.dk/			636537731089499543	

Add new map

JIRA project key	JIRAurl	TrimbleConnect PID	TrimbleConnect url	EasyAccess PID	EasyAccess url
	http://ic.novapoint.com:8080/ ▼		▼		▼

Submit

User Settings

Please fill in you username for JIRA, which can be found under profile.

For Connect, find your userId by logging into trimble connect, and then go to https://app.connect.trimble.com/tc/api/2.0/users/me (https://app.connect.trimble.com/tc/api/2.0/users/me). The "Id" value is your userId

To get the EasyAccess userId...

Currently registered users

JIRA	Connect	EasyAccess
aasmund.gjerde		636523882026584399
abfr		636379642046954348
abh		634670256603783558
adbr		636344205136219312
adri		636385643329460053
adsd		635642680918510614
anders.norheim		636282032565760232
aped		635766223723136957
arkn		636530752806504993
armin.lutz		636150152472356567
arve.krogseth		634843573263642028
asle.staland		636238886084763844
biatbg		635023203116030810
bjor		636425398821035434
brynjar.thorsby		634550505482547500
cbsn		636383843036645988
cgch		636468668292430937

trond.simensrud	636150566919688505
tscosl	636543036494031140
tslosl	634715754365573041
ulki	636402026927916927
vera	636384884904133394

Jira Username	TrimbleConnect Username	EasyAccess Username

Submit

Update DB (update)

Appendix J

Requirements for Connecting Jira to IssueConnector

Requirements for connecting JIRA to IssueConnector

By Øystein Bjerke Hauan

Api User

A dedicated user with full read, write, update, delete permissions on issues.
The user must also have access to set/edit reporter of issues.
The user must be able gain access via Basic Authentication.
The user *must* be an administrator (otherwise the comments will not work).

Fields

The issue types that are to be connected *MUST* (at this time) contain the following fields:
Standard fields:

- Summary
- Reporter
- Assignee
- Labels
- Status
- Priority

Custom fields:

For each connected system, there *must* be a field with that systems name and object model for its ID and URL. These should be standardized, but the current versions are:

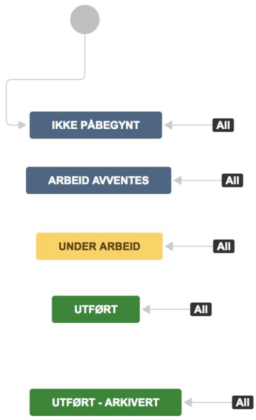
EasyAccess: EasyAccessTopicId, EasyAccessTopicUrl
Trimble Connect: TrimbleConnectTodoId, TrimbleConnectTodoUrl

Any other systems must follow this name convention.

All the above fields must be accessible in the create, view and edit screens for the API-user. Ideally, these should not be editable by any other users (except maybe project administrators). If any of these fields are manually edited or removed, the system will fail.

Status and Priority

The workflow in the project must match the statuses in the connected systems. The simplest way to achieve this is to create the statuses with exactly the same names, and allow the workflow to transition to each state without strict transition rules.



Above is an example, where all statuses can be set from all the others. Remember both the name of the status *AND* the transition names must match exactly.

The names of priorities must also match exactly with the connected systems for this to work properly.

Add-ons

Jira (for some reason) does not allow for the author of comments to be set by the api user (administrator) out of the box. There is an extension that allows for a workaround for this, that must be installed. It's called "Extender for Jira" by it.Lab Adam Labus.

Attachments

Attachments must be enabled.

Webhook

There must be a webhook which posts to the url of the IssueConnector. This can be configured in settings → system → webhooks. There should be one for issues which posts to <issueconnector-base-url>/api/jira/issues. Here one can also specify JQL's for which updates the IC should receive.

There should also be a webhook for the comments, which posts to <issueconnector-base-url>/api/jira/newcomment?commentId=${comment.id}&issueId=${issue.id}. The get parameters here can be configured in the setup of the webhook.

Users

As a current workaraound, all users from the JIRA system have to be mapped manually (only once) to work with the other systems connected with the IC. A list of users who are to be in both systems, should be sent to the administrator of the IC. (username in JIRA, userId in EasyAccess etc.)

Appendix K

Adding a New Service to the IssueConnector

Adding a new service to IssueConnector

By Øystein Bjerke Hauan

Introduction

This guide describes the process of adding a new service to the IssueConnector.

Cloning the repository

THE STRIKED STEPS ARE DEPRECATED AS OF 15.5.2018.

~~In order to clone the repository, you need access to the repo at https://bitbucket.trimble.tools/projects/IC/repos/issueconnectorwebservice. Contact the product owner to gain access.~~

~~You need to generate an ssh-key for bitbucket. Consult with the repo admin to do this.~~

~~Clone the repo to the desired location on your computer. (If you are new to Git, there are several great guides available online).~~

~~Open the project in Visual Studio, and from there you can setup the Team Explorer with the repo.~~

~~PS: This step will be unnecessary when the IssueConnectorLib is released as a nuget package (which it hopefully will be). Then you can create a new project in VS and just install this package to get the functionality you need.~~

Getting Started

1. ~~After the project is opened and indexed in visual studio, create a new branch for your work. For example, if you are adding a new service to the issue connector, name the branch *feature/<name_of_service>* and checkout the branch (from master).~~ **Create a new solution in Visual Studio.**

2. ~~In the Solution Explorer, once you are checked out to your new branch,~~ Right click the solution and add a new project. Add a new *.Net Core → ASP.NET Core Web Application*. If you wish to just create an API, choose Web API, otherwise choose Web Application (Model-View-Controller) (recommended) .Name your service appropriately (*<yoursystemname>Service*, e.g. *EasyAccessService*).
3. Now you have a new service, with a few things generated for you. Depending on which type you choose in the previous step, you will have three folders (models, views, controllers) or just controllers. In addition you have two cs classes, *Program.cs* and *Startup.cs*. The latter is the only one important now. In addition you have an *appsettings.json*, which you will need later.
4. ~~Add a reference from *IssueConnectorLib* to your new project by right clicking *<yoursystemname>Service* and selecting *Add → Reference → IssueConnectorLib*.~~
5. Install the NuGet package *IssueConnector* by *oysteinbhauan*
6. Now you are ready to start!

Connecting to your system

The first thing we need to do is to authenticate the IssueConnector to connect to the system you want to add. There is no standard way to do this, but generally you will have to add a specific user to this system that the issue connector can use. Depending on the authentication your system uses, you will need to adhere to this in the code.

1. Create a new folder called *Client*.
2. Add a new class called *<yoursystemname>ResourceClient.cs* (Example JiraResourceClient.cs or EasyAccessResourceClient.cs)
3. Implement the interface *IResourceClient<T>* where T is the type of client you use to connect. For example, EasyAccessService uses a standard HttpClient (System.Net.Http) while Jira uses a framework.
    a. Example : public class EasyAccessHttpClient : IResourceClient<HttpClient>
4. This interface includes a couple of methods required to connect to the system. Just leave them empty for the moment.

Now we need to add the functionality for the service to connect and gain access to the API of your resource.

1. Add a new folder called *Connector* with a new class called *<yoursystemname>Connector.cs* (Example EasyAccessConnector.cs).
2. Here you need to add at least one method (e.gtConnect) that returns the type T (or Task<T>) of your resource client. This should ideally be a static method. (For inspiration on connecting Trimble systems, look at TrimbleEasyAccessConnector.cs, which uses the NuGet package Trimble.Identity to get the right tokens.)

3. At the moment you can just hard code the urls and usernames and passwords. We will emigrate these later.

Now it is time to initialize the resource client.

1. Go back to the <yoursystemname>ResourceClient.cs. In the *Connect()* method, set the RestClient to <yoursystemname>Connector.GetConnectedClient.
    a. RestClient = await TrimbleEasyAccessConnector.Connect(<connection parameters>); (The parameters here are the required params for connecting to the system).
2. If you need any special logic for reconnecting your resource, you can add that as well.

Now we need to inject this client into the application. If you are unfamiliar with this approach, read up on the basics of .Net Core 2.0.

1. Open Startup.cs
2. In the ConfigureServices(IServiceCollection services) method, add the following:
    a. services.AddSingleton(IResourceClient<T>, <yoursystemname>ResourceClient>();
        i. Here the type T is again your client, for example HttpClient.
3. In the Configure method, add the following line:
    a. app.ApplicationServices.GetService<IResourceClient<T>>().Connect().Wait();
    b. This connects the system and waits for it to be connected before starting up.

Now you should have a working ResourceClient which you can use for the CrudServices!

Creating the CRUD service

Now it is time to create our crud service.

1. Create a new folder called Services and add a class named <yoursystemname>CrudService.cs.
2. Implement the interface IResourceCrudService. Now you will have a lot of methods that you may or may not want to implement. Just focus on the ones you need, the others can be left empty.
3. Generate a constructor which injects the IResourceClient<T> to private field _resourceClient.
4. Inject this in the application by adding
    a. services.AddSingleton<IResourceCrudService, <yoursystemname>CrudService>(); to your startup
    b.

5. Now you are ready to start doing crud operations on your client. You can test this now by going into ReadInstanceAsync and testing with the different methods of _resourceClient.Client. Check out previously implemented services for examples.

Injecting the DatabaseService

We need to access the issueConnectorDatabase. This can only be done at Trimbles location (at the moment). The database is where we find user mappings and project mappings.

1. Open appsettings.json and add the following: "ConnectionStrings": { "DefaultConnection": "server=80.232.45.109;user id=jirauser;password=test123;port=3306;database=issueconnectordb;" } (Current setup)
2. Open Startup.cs and add the following lines:
    a. services.AddTransient<IDatabaseService, DatabaseService>();
    b. services.Configure<DatabaseConnectionString>(Configuration.GetSection("ConnectionStrings"));
3. This will give you access to the basic databaseMethods found in DatabaseService (IssueConnector). Inject this in your constructor in the CrudService we created to private field _dbService.

If you need more specialised db methods, generate a class that inherits from DatabaseService and create your own interface. Check out EasyAccessDatabaseService for an example.

Adding a publisher

Install the follwoing NuGet packages: MassTransit, MassTransit.RabbitMq, MassTransit.Log4Net (if you are yousing Azure, MassTransit.AzureServiceBus).

Our publisher is going to handle the outgoing messages in our service. It will need to inject the CRUD-service.

1. Create a new folder called Services and add a Class called <yoursystemname>Publisher.cs
2. Implement the interface IResourcePublisher from IssueConnectorLib
3. Create a constructor and inject the CRUDService, as well as any IOptions you may require.
4. Generate the following method:
    a. private async Task PublishInstance<T>(T instance) where T : class { }

   b. Add the following lines to the method (this must be changed when moving to azure service bus):
      i. var bus = ServiceBusProvider.GetRabbitMqPublisherBus(<RABBITMQURI>);
      ii. var busHandler = await bus.StartAsync();
      iii. await bus.Publish<T>(instance);
      iv. await busHandler.StopAsync();
5. Inject the publisher into your application by adding to your ConfigureServices in Startup.cs:
   a. services.AddTransient<IResourcePublisher, <yoursystemname>Publisher>();
   b.

A new bus will be created for every outgoing message to ensure independent handling of the messages. We will get back to the publisher after adding the event listener.

Adding the message consumer

Now it is time to add the consumer, or subscriber, which will listen to all messages of the desired types that go through our message bus, and translate these in to objects that we can work with. All this is done automagically through the MassTransit open source SDK (http://masstransit-project.com/MassTransit/). Please reade the documentation for more information on how to handle the messages. The documentation is great and simple for the most part.

1. In the services folder, add a class <yoursystemname>Consumer.cs
2. Implement the interface IResourceConsumer from IssueConnector
   a. This interface includes methods for all types of contracts, IStandardInstance, IStandardViewpoint, IStandardComment etc. You need only implement the methods of the contracts you are interested in.
   b. It also includes some methods for handling the messages, but only for some, as this is a work in progress.
3. Create a constructor that injects the CRUDService, the resourcePublisher, and any IOptions you may need.
4. Add the private field IBusControl _bus;
5. In the method StartConsumerAsync() add the following lines (Also remember to include any relevant logging)
   a. _bus = ServiceBusProvider.GetRabbitMqConsumerBus(IResourceConsumer consumer, string uri, string systemName, string systemId);
      i. The 4 params here must adhere to the following:
         1. Use **this** resource consumer to create the queue.
         2. The uri of the rabbitMQ. This will be ”rabbitmq://localhost” if it is on the localhost. This should be moved to the configSettings.

3. The name of the system, e.g Jira, EasyAccess, TrimbleConnect etc.
4. A unique ID for the system. This must be the same every time the system is restarted.

ii. You will have something like this: ServiceBusProvider.GetRabbitMqConsumerBus(**this**, **_configSettings.RabbitMqUri**, **_configSettings.SystemName**, **MachineInfo.GetMacAddress()**);

b. await _bus.StartAsync();

6. In StopConusmerAsync add await _bus.StopAsync();

In each of the implemented Consume methods you can handle the incoming messages of different types. Access the incoming object by using context.Message.
This is also where you should apply a filter to filter out messages you dont want, and also any decryptors that should be implemented at some point (but not yet).

Remember to inject this in your startup class.

Also add app.ApplicationServices.GetService<IResourceConsumer>().StartConsumerAsync(); to the configure method in startup.cs

Adding an Event Listener

This is where it might become tricky. We wish to be able to sit on the fence and observe what happens in our external system, but this might not be as easy as one would hope. In JIRA, you can define webhooks, which sends a POST request to specified url when something of interest happens in the system, e.g an issue is created or updated, or a comment is added. This makes it very simple, as you can also define *when* the webhook will send requests, so you eliminate unnecessary messages.

EasyAccess does not have this functionality. Instead it uses a SignalR event notifier, which is not that configurable. We define which project we want to listen to, and then receive every event in that project, so we must have stricter filters in the service.

For your system, you need to explore what the possibilities are. Many webservices have some sort of eventlistener implemented (at least they should have), and hopefully it will be possible to connect to it. This will be different for almost all systems, therefore it is difficult to create a general way of doing this. In JiraService it is done through a controller (API) and in easyaccess through SignalR NuGet-package.

When you have figured out how to listen to events from your system, create a folder in the project called EventListener. (*If you are listening using a webhook, it is easier to generate an API in the Controllers folder.*) Create an interface and a class for your listener, e.g. IEventListener and <yoursystemname>EventListener. If the listener has to be started explicitly, you should include a *StartAsync()* and *StopAsync()* method in the interface, so it can be started from the *Startup.cs* class. The eventlistener should inject the *IResourcePublisher* and any other dependencies you may need. For inspiration on connecting to a SignalR client, check out the EasyAccessService EventListeners.

It is important to implement a filter to contorl which events are published to the IssueConnector. For example, if a new issue is created by your service because of a new instance received through the issueconnector, it is important that your service does not post this, but only its url and identifiers through an IStandardIdentifierMap.

Handling messages

Now you should have two-way message flow both to and from your connected system. It is time to start handling the messages that go out and come in.

Adding a message filter

In a publish-subscribe pattern, it is not necessarily a guarantee that all messages our service receives, will be of interest. Therefore, it is vital to implement a filter that reads the messages and decides if it should be processed or discarded. These criteria must be individually configured to suit your needs, but it is advised to use deoendency injection for this as well.

Mapping objects

This is one of the main challenges with integration systems. You need to find a way to map objects to suit your system. The IssueConnector uses standard contracts, IStandardInstance, IStandardComment, IStandardViewpoint etc. Messages coming from the IC will always be of this type, and all messages published for your service must be of these types.

Appendix L

Challenges and Future Work for the IssueConnector

Challenges with IssueConnector and Future Work

by Øystein Bjerke Hauan

This document presents and discusses some of the challenges with the IssueConnector (IC) that need to be resolved. These are what I, the initial developer, would describe as crucial if the product is to be released to the public. Some of these are issues I didn't have time, or resources to implement, as the implementation period was quite short and hectic.

Deploying services on HTTPS protocol

The IssueConnector (and self hosted Jira) is running over HTTP, which is very unsecure. This should as soon as possible be set up with SSL/HTTPS. The process for doing this for IIS applications is described here.

(https://support.microsoft.com/en-us/help/324069/how-to-set-up-an-https-service-in-iis)

A proper database

At the time this document was written, the IC uses a MySQL Community database to store some simple data. This includes a project map, a user map, EasyAccess user information. The latter can be rewritten to fetch user information from the EasyAccess api for Topic Project Settings (https://topics.quadridcm.com/Help/Api/GET-api-projects-id-topicprojectsettings) and mapping the username to firstname and email, and thereby removed from the database (OR the EA API could support just using the username, and setting the correct fields based on that).

It should also be discussed wether or not there should be an instance map in the database, for each issue. At the moment, these maps are stored at each system. In Jira, this is not a problem, as you can just add custom fields to contain this information. But not all systems have this opportunity, so in EA it is stored in the comments, and in Trimble Connect in the bottom of the descritption. The problem with this approach is that the size of the database can potentially become enormous - but it may not be a problem. It should at least be discussed.

The JIRA database for the current installation is also in this MySQL community edition. This can be exported easily thorugh JIRA to an XML-format and moved to a new database.

The database should probably also be split up into different customers, so that project and user maps are not shared the way they are now.

User Interface

The IC is primarily a backend system that requires little interaction once it has been configured. But this configuration service must be properly implemented to support mapping of fields, database interaction, adding project and user maps etc. I have started with this in the project called IssueConnectorConfig in the IssueConnectorWebService solution in bitbucket, but it has a long way to go.

This IssueConnectorConfig is thought to host a web service for adminstrators of *the entire system*, so very few should have this access. In addition to this, each of the services should have a config page where the administrators of that particular service and projects can map fields, users, etc. I have made several preparations for this. If you use ASP.NET Core Web Application when designing new services, you will get all this set up and ready to go when you start.

Splitting the code up in to different solutions.

At this time, all the code is located in the IssueConnectorWebService repo. It does not need to be this way. After the refactoring of the architecture, I have made a project called IssueConnectorLib which contains most of what a service will need to implement. This Lib is released as a NuGet package (called IssueConnector by oysteinbhauan), so that “anyone” can make a new service to attach to the IssueConnecor. This package contains interfaces for connecting to and communicating through the message bus, as well as optional interfaces for setting up the crudservice, client connectors, objectmappers etc.

Encryption of messages

Most service bus providers don’t offer encryption of messages, because it doesn’t really make sense. What you would want is a standard end to end encryption, which means that only the publisher and the correct subscribers can read the message. This is also a way to ensure that messages don’t end up in the wrong services, if all messages are to go through the same message bus. If a service cant decrypt a message, it should simply ignore it. Since all services subscribe to the same *IStandardIntance* contract, every service connected to the service will receive every message that goes through the Bus, sparking the necessity of encrypted messaging.

Clustering

As a way to guarantee that all messages are handled correctly and are not picked up by the wrong receiver, a solution could be to cluster the services, depending on the customer. This means adding a separate service *for every connected system* and connecting them via a dedicated message/service bus. This is probably the best way to ensure customers that their data will not end up in the wrong hands. There are a lot of sensitive, personal data being sent through the IC, which makes it crucial to ensure that these are not misplaced.

Filtering and Authentication

The filters in the IssueConnector are at the moment very simple. The filters need to applied both ways, for incoming messages as well as outgoing. Each service needs to specify requirements for the messages they receive. Depending on how the IC is deployed, implementing two way authentication should be considered. At the moment, the only authentication needed is between a service and its system, but one could imagine that attackers could target the IssueConnector directly.

Cloud Deployment vs Self Hosted

At the moment, the IssueConnector is running out of Visual Studio on a dedicated server. There have been expressed wishes to deploy the IssueConnector to Azure. Howver, if the IssueConnector is going to be distributed as a package to customers with each customer using its own cluster, (which I beleive would be the way to go) this can become cumbersome and expensive. A better idea could be to host it at Trimble, and deploy the IssueConnectors in virtual machines for each customers, e.g. Docker, OpenShift etc. This requires skilfull deployment, but it will probably be a worthwhile investment. This way, one ensures 100% separation of messages between customers, and complete independence between these.