



# An integrated software platform for the efficient management of projects

**J Botma**

 [orcid.org/0000-0002-2399-6459](https://orcid.org/0000-0002-2399-6459)

Dissertation submitted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor: Dr JF van Rensburg

Graduation: October 2018

Student number: 29891027

# Abstract

**Title:** An integrated software platform for the efficient management of projects  
**Author:** J Botma  
**Supervisor:** Dr Johann van Rensburg  
**School:** North-West University Potchefstroom Campus  
**Degree:** Masters in Computer and Electronic Engineering

Electronic software platforms play a role in the internal processes and management of project-based firms. Many task-based solutions exist in practice to handle projects efficiently. These tools address shortcomings that prohibit the effective management of projects.

This study identifies a subset of problem areas that task tools address. Further investigation showed that integration, planning, communication, record keeping and knowledge transfer all improve the project. Each of the mentioned concepts influences the way in which projects are managed efficiently.

Several design patterns and software development workflows were investigated to build a concrete software platform with multiple business applications for one of which is a primary project task tool. A combination of the iterative and agile development workflow suited the development best. The software platform was designed from the needs and requirements that emanated from the literature study.

Functional verification proved that the task tool was effective, and several statistics showed that the tool is being used actively for multiple purposes. A survey validated the business- and user requirements for the tool and software platform. The approval rating yielded on average that the business- and user requirements rated above 4.14 out of 5 on average for all the project requirements.

Users agreed that the implemented task tool is integrated and addresses all the concerns from the study regarding the efficient management of projects. 70% use the platform to manage their projects and above 80% of the users also use the software platform for IT and development support.

---

**Keywords:** Tasks, project, software platform, management,  
efficient, projects, record keeping, knowledge transfer

# Acknowledgements

I would sincerely like to thank:

- My parents for their love and support during tough times.
- Dr J. van Rensburg, Dr M. Mathews and Dr J.N. Du Plessis for their guidance and insights.
- Prof. E.H. Mathews and Prof. M. Kleingeld, for presenting an opportunity for studying at CRCED Pretoria.
- TEMM International (Pty) Ltd for sponsoring and facilitating the research process for the study.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Abbreviations and nomenclature</b>	<b>vi</b>
<b>List of figures and tables</b>	<b>vii</b>
<b>1 Introduction and background</b>	<b>1</b>
1.1 Preamble . . . . .	2
1.2 Efficient management of projects . . . . .	2
1.3 Key concepts that improve efficiency in projects . . . . .	3
1.4 Existing solutions . . . . .	9
1.5 Problem statement . . . . .	13
1.6 Need for the study . . . . .	14
1.7 Research process and implementation . . . . .	15
1.8 Chapter overview . . . . .	15
<b>2 Background of efficient management and software solutions</b>	<b>17</b>
2.1 Preamble . . . . .	18
2.2 Time efficiency and project management . . . . .	18
2.3 Software development workflow models . . . . .	24
2.4 Programming languages for web-development . . . . .	28
2.5 Web-based design and program architecture . . . . .	30
2.6 Literature objectives . . . . .	33
2.7 Conclusion . . . . .	35
<b>3 Design, development and deployment of a software platform</b>	<b>36</b>
3.1 Preamble . . . . .	37
3.2 Processes and models . . . . .	37

---

3.3	Requirements . . . . .	38
3.4	Technical design . . . . .	41
3.5	Testing and user training . . . . .	53
3.6	Conclusion . . . . .	54
<b>4</b>	<b>Results</b>	<b>55</b>
4.1	Preamble . . . . .	56
4.2	Software platform . . . . .	56
4.3	Project task tool . . . . .	57
4.4	Verification . . . . .	62
4.5	Validation: Case studies and survey . . . . .	64
4.6	Quantative results and general statistics . . . . .	72
4.7	Summary of statistical results . . . . .	75
4.8	Conclusion . . . . .	76
<b>5</b>	<b>Conclusion</b>	<b>77</b>
5.1	Discussion and conclusion . . . . .	78
5.2	Recommendations . . . . .	79
	<b>Bibliography</b>	<b>80</b>
	<b>Appendix I Project task tool survey</b>	<b>89</b>
	<b>Appendix II SQL queries, calculations and results</b>	<b>103</b>
	<b>Appendix III Tasks Manual</b>	<b>110</b>

# Abbreviations and nomenclature

API	Application Programming Interface
CSS	Cascading Style Sheet
DAL	Data Access Layer
DB	Database
DOM	Document Object Model
HRD	Human Resource Development
HRM	Human Resource Management
HTML	Hypertext Markup Language
IT	Information Technology
JSON	JavaScript Object Notation
MVC	Model View Controller
MVP	Model View Presenter
MVVM	Model View ViewModel
ORM	Object Relational Mapping
PM	Project Management
PMP	Project Management Plan
RAD	Rapid Application Development
ROI	Return On Investment
SQL	Structured Query Language
URL	Uniform Resource Locator
WBS	Work Breakdown Structure

# List of Figures

- 1.1 Industry solution to project efficiency management . . . . . 3
  
- 2.1 Functional diagram of the waterfall model . . . . . 25
- 2.2 Functional diagram of the iterative model . . . . . 26
- 2.3 Functional diagram of the prototyping development model . . . . . 27
- 2.4 Functional diagram of the agile development model . . . . . 28
- 2.5 Components of the MVC pattern . . . . . 31
- 2.6 Components of the MVP pattern . . . . . 32
- 2.7 Components of the MVVM pattern . . . . . 33
  
- 3.1 Steps followed in the creation of the software platform . . . . . 38
- 3.2 Task with different types of data that can be linked to it . . . . . 40
- 3.3 Functional framework block diagram . . . . . 42
- 3.4 Back end functional flow . . . . . 42
- 3.5 Object relational mapping . . . . . 44
- 3.6 Adapted MVC pattern . . . . . 45
- 3.7 Functional system flow block diagram . . . . . 45
- 3.8 Create or update task block diagram . . . . . 46
- 3.9 Example of a JSON key-value pair object . . . . . 47
- 3.10 Routing block diagram . . . . . 47
- 3.11 Front end browser block diagram . . . . . 48
- 3.12 Front end flow block diagram . . . . . 49
- 3.13 Task list design diagram . . . . . 50
- 3.14 Create task design diagram . . . . . 51
- 3.15 Task feed task design diagram . . . . . 52
  
- 4.1 Software platform tools . . . . . 56
- 4.2 Task list . . . . . 57
- 4.3 Create and edit task interface . . . . . 58
- 4.4 Configuration of a recurring task . . . . . 60

---

4.5	Task Feed Interface . . . . .	61
4.6	Percentage of each task type . . . . .	65
4.7	Spread of employees in different groups . . . . .	69
4.8	Percentage of people working in different groups using the task tool . . . . .	69
4.9	Average response of user agreement with functionality of the tool . . . . .	71
4.10	Creation of tasks over the months since platform deployment . . . . .	72

# List of Tables

- 1.1 Comparison of existing task-based project management tools. . . . . 12
- 2.1 Types of software development workflows . . . . . 24
- 2.2 Comparison of programming langauges with respect to web development . . 29
- 4.1 Verification of business requirements . . . . . 62
- 4.2 Verification of user requirements . . . . . 62
- 4.3 Verification of functional requirements . . . . . 63
- 4.4 Number of tasks linked to the different types of labels . . . . . 73
- 4.5 Descriptions and values of the different statistics in the chapter . . . . . 75

# CHAPTER 1

---

## Introduction and background

---

*'I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me'* - Isaac Newton

## 1.1 Preamble

This section contains an introduction to the efficient management of projects through the use of task tools. Key areas were identified from literature that describes the factors that influence the success of a project. A comprehensive summary of task tools and various task management tools are compared to identify problems in industry. A discussion on how these problems will be addressed will follow.

## 1.2 Efficient management of projects

Companies undertake projects to generate income [1]. A project is considered to be successful if the outcome meets all of the expected requirements and was profitable [2]. A successful project requires efficient management to maximise profit [3].

*Efficiency* is perceived as a percentage of improvement on a previous measurable state.<sup>1</sup>  
<sup>2</sup> <sup>3</sup> Efficiency, from an engineering perspective, is applicable *after* the system has been implemented due to physical outcomes emanating from work. Literature on efficiency in project management also has an outcome-based focus [4].

*Management* is an umbrella term that contains human resource-, project- and process management [5, 6]. Proper management requires a company to identify the characteristics that sustain positive outcomes and to evaluate the efficiency of a project internally and externally. The company also needs to recognise situations that pose a potential threat to the project outcome.

Many approaches exist to manage, examine and evaluate a company and its resources [7], such as task- or project-based solutions. These approaches focus on outcomes that strive towards the success of the project. General project management includes processes and outcomes that do not necessarily focus on the internal efficiency of the process or company itself [2]. Task-based tools can potentially improve the efficient completion of projects.

Figure 1.1 gives an example of the use of a task-based management tool [8, 9]. It shows the procedure followed for project management through the use of such a tool. The project procedure is broken down into several tasks during the planning phase before project im-

---

<sup>1</sup> <https://en.oxforddictionaries.com/definition/efficient>

<sup>2</sup> <https://en.oxforddictionaries.com/definition/efficiency>

<sup>3</sup> <http://www.dictionary.com/browse/efficiency>

plementation. Users work individually or in groups to complete tasks. During all phases of project implementation, a web platform aids by providing feedback on the state of the project. This feedback encompasses communication, progress and categorisation of tasks to promote the *efficient* management of the project.

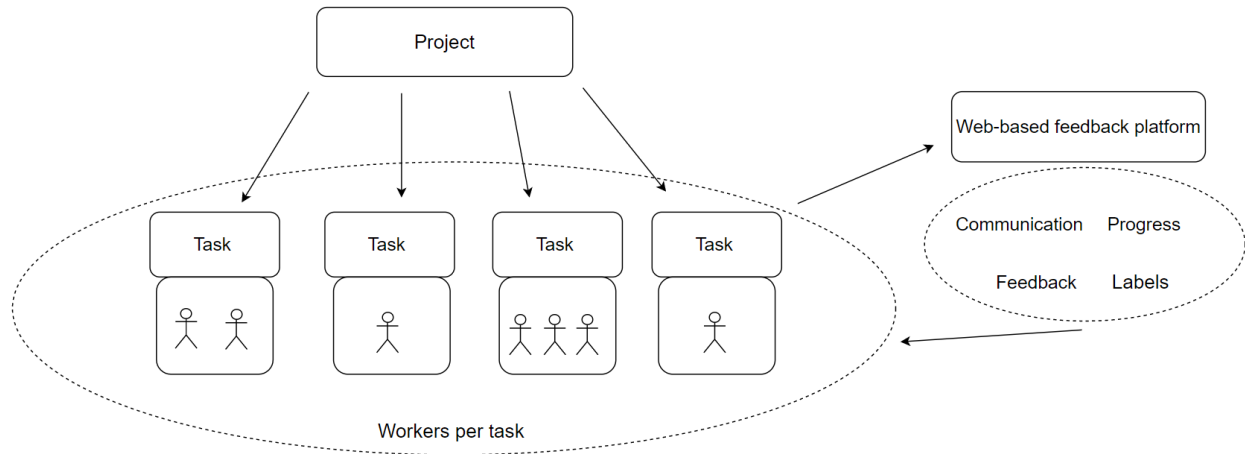


Figure 1.1: Industry solution to project efficiency management

Project-based companies use task tools, like in Figure 1.1, and have numerous projects running at once. The companies use these projects and their respective outcomes to manage project performance [10]. Projects gain from improving the efficiency of operations and saving resources [1]. Throughout all phases of the work process and the development of the solution, internal efficiency should be taken into consideration to enhance profitability.

This study focuses on a case study of a specific company seeking to enhance its profitability through the efficient management of their projects. This will be done through design and implementation of a task-based tool and integrating it with an existing web-platform. A software platform that provides a task tool would give the company a competitive edge.

### 1.3 Key concepts that improve efficiency in projects

Efficient management of projects has a wide scope. The subject is applied to several academic areas, such as project management [11], risk reduction [12] and even project planning [5]. Literature describes task-based organisational tools to help improve on several areas. The areas which affect task-based tools are described below [6, 4]:

- **Efficient use of human resource management** : This describes human resource management (HRM) with relation to the group dynamic, and personal and group motivation. A task tool links users to collaborate on sections of a project.

- **Improved process efficiency** : Project management focusses on communication and centralised knowledge influences efficiency. Current processes within companies and their pitfalls are identified and managed through a task tool.
- **Better knowledge transfer** : Information and knowledge are generated within an organisation. It will investigate which mechanisms are used to generate knowledge, and thus how knowledge is transferred from employer to employee [13]. Task-tools may be used to identify important processes in a project.
- **Record keeping** : This is a key mechanism from which knowledge generation occurs in a company. This describes the importance of efficiency as the structural information foundation of growing organisations. A task tool saves all information relating to the users doing the work.
- **Project planning** : Planning minimises risk [14]. A task tool allows a project to be broken down into smaller sequential steps. It allows the user to have a better view of the project scope.

### 1.3.1 Group efficiency and human resource management

The first dynamic in the success and efficiency of a project is described in this section. It lies in the people and how they work together [15]. The success of a project may be correlated with internal factors, such as group efficiency. A typical project in an organisation requires group work [16].

Human resource management is used in multiple contexts to describe how people should be organised and handled in a company. HRM creates a group dynamic that influences how people perform and relate in the workplace. This dynamic is dependent on a variety of factors, such as [17]:

- Individual age
- Cultural differences
- Level of competency or education
- Beliefs
- Personality

Most of these factors cannot be measured discreetly. How they influence each other is also not tangibly quantifiable. Studies show that they still affect each other [18, 19]. A hypothesis is made that the grouping of individuals has a impact on the efficiency of the workload.

Grouping needs management, and studies show that people who have different views on work processes may become unhappy if placed together [20]. How the group drives towards a common goal has an effect on the amount of time needed, as well as the quality of the end product.

Motivation forms a large part of effective management within the dynamic in a group [20]. It is primarily divided into personal motivation and group motivation. Personal motivation comprises a personal belief system, while group motivation flows from a common view. Personal motivation follows the influence of the group motivation.

People have a definitive correlation to the efficiency of the project [21, 2]. Task-based tools serve well in tackling people management head-on by providing a communication mechanism. These tools also help to group workers together and delegate work to the parties involved. It can be concluded that several factors that lie within the HRM of project and company influence the success and quality of the project by using the topics discussed in this section.

The term ‘human resource management’ stems from Harvard and can be described as ‘management decisions that affect the relationship between the organisation and employee.’ [16]. Human resource development (HRD) describes the investment a company can make in its’ workers to help them grow.

Gorman et al. discuss the relevance of HRM in project management [18]. The findings suggest that many project management techniques exist, but that little emphasis is placed on *how* the project is implemented. The human aspect of a project contains subjective traits, such as:

- User feedback within a company between the employer and employee.
- Training and competency of the workers within a company.
- Fairness of the organisational procedures and structures.

These traits influence how the personnel learn and perform their day-to-day work. The development process is part of the institution, for example in universities. The lecturers stay in the academia to gain intellectual fulfilment [22]. In practice, job satisfaction and personal growth lead to efficient workers [23]. Better workers lead to improved project efficiency.

An important trait to note is that the competency and development of the staff are essential to the employer as well as the employee [5]. Record keeping and knowledge transfer, discussed later, are concepts that exist as a part of a organisation to develop the workforce.

### 1.3.2 Process and project efficiency

The processes and activities that take place for the duration of a project define the outcome within an organisation [24]. When the processes are monitored and evaluated, continuous improvement can become part of the building blocks that form a successful company [24].

A process is defined as a ‘series of actions or steps taken to achieve a particular outcome’<sup>4</sup>. A process, or mechanism, is designed to achieve an expectation. If it fulfils this expectation, which contains several areas, the project is considered to be complete and a success [25].

This is, however, not the case, since the efficiency of the expectations is often overlooked. The situation begs the question: ‘Does the end justify the means?’ In short: No, not when it comes to business or people.

Negligence in internal process quality or efficiency is seen in the structures of most work processes or development models. This is why quality control applies to an important individual step [26]. The improvement is only part of the *final* steps of these models, whereas literature suggests that it should be done *during* the project process [27, 28].

Iterative models [27] such as in continuous improvement, are seen as the end point or, last step. A safe assumption is that, without continuous improvement, the project is doomed to degradation or failure. The ‘project’ is interpreted as ‘an enterprise that is carefully planned to achieve a particular aim’<sup>5</sup>.

Measurement and comparison of work can prove to be difficult due to the ever-evolving nature of a project that generates scope creep [29]. Scope creep refers to the changing nature of requirements that prolong the project lifetime. Proper planning and risk management are needed to improve the efficiency of a project [30].

The abovementioned implies that there are many factors that contribute to the process and project efficiency, discussed in Chapter 2. Time efficiency, through planning and risk

---

<sup>4</sup> <https://en.oxforddictionaries.com/definition/process>

<sup>5</sup> <https://en.oxforddictionaries.com/definition/project>

management, and proper processes, through continuous improvement thus play a key role in the outcome of a project. The only way to keep track and evaluate the progress of a project is through record keeping.

### 1.3.3 Record keeping

Documentation plays an important part in the success of maintainability and accountability [31]. Record keeping of procedures is a tool that retains knowledge and is an important mechanism used in knowledge transfer [32]. The accountability acquired by linking the employer to his work, assuming a paper-trail, results in a more responsible employee.

With a full record it is possible to trace how or what has been done to perform maintenance. Documentation of procedures during the initial phases of a project provides a framework for later decisions [6, 15].

When record keeping is part of the entire process, the documentation, especially in code, may prove later to be invaluable to the project. Thus, to derive value from a project in a company, record has to be kept to enable knowledge transfer and retain valuable information about the project [15].

Often the record-keeping process is time consuming and tedious. J. L. Cummings and B. Teng describe the key variables regarding knowledge transfer in an organisation [33]. A problem that arises in practice is the complex structure of the knowledge accumulated through the lifetime of an organisation.

This can be embedded in the people and their skills, the technical tools and the systems that form part of the company. The paper argues that it may be difficult to retain and recreate the knowledge. To separate learned knowledge and generate a reservoir of information in the company will enable it to stay competitive [13].

According to L. Argote and P. Ingram, tasks ‘reflect the organisation’s goals, intentions and purposes’. The article suggests that these tasks form a network of information. Specified tools exist to retain the comprehensive information, but a general task tool may serve a better purpose.

### 1.3.4 Knowledge transfer

Employees in an organisation gain knowledge through day-to-day work and this knowledge does not only include the how-to, but also what to do [15].

B Muskat and M Deery suggest that employees work in different teams and constantly move from positions [15]. The knowledge gained from a new employee in his previous team has to be acquired again once the old employee has moved from his position. They describe how the organisation has a memory and gains from retaining experiences. With a system in place to inform a worker how and what should be done, project efficiency can increase.

Knowledge transfer has the goal of retaining information in the company [15], and requires a fertile work environment to grow [34]. The employee has not only to learn actively, but the company needs to have processes in place to accommodate knowledge transfer. This is where user collaboration is important.

If strong communication is available in a company, employees will be able to learn quickly. The knowledge retained in a company improves the efficiency and likelihood of success in future projects [35], and is therefore is an important aspect of efficient management. Chapter 2 takes a closer look at what the real world industry is, with existing solutions to see how efficient management is supervised.

The complexity of the project varies through its life cycle [36] and the only way to retain the experience is through having company procedures in place. Knowledge is not only gained through experience, but also through the act of active learning. A study shows that one of the main factors that prohibit learning is the organisational culture [6]. Knowledge transfer describes the process in which mastery of a subject is transferred among entities in an organisation. For a company to remain competitive, it has to have processes in place to actively retain and use the knowledge gained over a time span [34].

According to Huber: ‘An entity learns if through processing information the range of its potential behaviours is changed’ [37]. This means that an active effort has to be made by the individual and by the company to ensure that knowledge is transferred to the company. There remains some responsibility on the side of the individual. Factors such as motivation, seniority and willingness to learn to play a leading role in the ability for knowledge transfer [34].

### 1.3.5 Project planning

Planning can be achieved through the use of certain tools. The software platform for the study provides several tools or mechanisms to incorporate project planning. For planning to be effective and the project to have a high chance of success, stakeholders previously gained involvement even at resource level [38]. A project management plan is then developed to specify goals, scope, or even costs, on the current project. A project management plan is ‘a formal, approved document that defines how project activities are to be executed, monitored, controlled, and concluded’ [12]. The project plan can be broken down into:

1. Analysis
2. Design
3. Development
4. Integration and system test
5. System openness
6. Acceptance test

This plan may vary, depending on the application of the project. Chapter 2 takes a deeper look into how planning is done through the use of software tools.

## 1.4 Existing solutions

Previous sections conclude that multiple needs exist to manage the efficiency of a project, to supervise the workers and to profit from work through knowledge transfer. These areas of importance have existed since the dawn of project-based companies, and many real-world solutions have been proposed. In the world of today, a project management solution platform, or a task tool, provides the answer that the industry desires.

A project management solution focuses on the planning, resource allocation and risk identification beforehand. This type of tool is used to create a framework to be followed by the employees. A perfect example of a project solution would be ‘Microsoft Project’ . A task tool is different from a project solution. It also allows for planning, but in a less explicit way. This tool focuses more on communication and management during project implementation.

Features included in a task platform allow for progress tracking, task labelling and collaboration among users. Granular work can then be categorised, prioritised and easily managed. Many variances exist in companies that provide project/task management software [39]. These are discussed in the sections below.

### 1.4.1 Office tools for project management

Generic software do exist that can be innovatively manipulated by users to obtain certain outcomes, like task managers, contrary to the intent or design of the tool. Two examples include Outlook and Excel.

#### Outlook

Outlook is an email client built by Microsoft and is primarily used as a task manager. Any other email client, like Gmail or Thunderbird, would still suffice for this use.

J Gwizdka and M Chignell describe the pending email in an inbox with respect to a task-based user interface [40]. Each email represents a task, or contains, work for the employee. The task management process begins when an email is sent from one user to the other, where it ends up in an inbox. From this inbox the mail is moved to different folders. The unhandled mail lies dormant and may get lost over time. This software describes a ‘what you see is what you get’ user interface. Communication is the main focus, but time-management tools via calendar are also provided.

Some email clients contain a task manager. The basic functionality of creating tasks with due dates is available. People inform another by assigning tasks. The system sounds flawless in overview, but the foundation, a mailing service, contains flaws when it comes to a practical implementation.

The platform is created to send mail, not manage tasks. Tasks are sent, or users are notified, to users via the email client, not a dedicated platform. Tasks are stored locally and are not implicitly distributed to the users who need to do them. One-way messages are pushed through a system without any type of feedback from other users.

Problems like record keeping, organisation and syncing progress may pose a proportional threat due to the possible stagnating nature of a mail inbox. Other issues also arise with the responsibility that now lies with the user to keep track of messages. The organisation of such can become difficult, especially with limited prioritisation and categorisation functionality.

## Excel

Excel and other variants of spreadsheet software are used to make, amongst other uses, different lists [41]. These lists would be compiled for tasks, time values and deadlines. <sup>6</sup> A framework or layout of milestones and objectives is created in the software to plan a project. Features of this type of software include customisation and a powerful planning platform.

The user is free to create any planning and delegate work. This is very time consuming, and after the sheet creation, it stays static over the course of the project. There is no instrument to allow collaborative project tracking or communication through the use of this specific software. The software is independent of any other software system, which would make it difficult to integrate into a web based platform.

Lastly, additional software is needed to distribute the project plan and progress updates amongst users, since no provision for communication is facilitated. This is where task-based handlers come in. These tools are located in the cloud that provides a central hub for the current project that anyone can access and contribute to.

### 1.4.2 Generic task-based handlers

Task-based tools are the industry solution to efficiency management [39, 42]. This type of software solves problems that can occur when attempting to improve project efficiency.

Communication, knowledge transfer, project management and the grouping aspect of human resource management are all addressed through the use of the software. This set of tools is web-based and allows a central place for accessing and managing workflow. Following is a summary of a list of popular software available today [43]:

---

<sup>6</sup> <https://www.officetimeline.com/project-management/excel>

Table 1.1: Comparison of existing task-based project management tools.

Task Tool	Description	Advantages	Disadvantages
Trello [8]	Work collaboratively in an online platform that enables project management and efficiency.	Communication is a key selling point. Tasks are categorised via a card interface. A web-based tool that is accessible anywhere. People are coordinated into teams to work together.	Insufficient integration with external parties. No proper user interface feedback for completed tasks. Free package is limited; if full functionality is required, additional costs are incurred.
Asana <sup>7</sup>	Online task management software that supports project management and agile development.	Dashboard with immediate project feedback. Progress tracking and status available by tagging tasks with labels. Hour tracking linked to a task. Has a familiar social media feel.	Insufficient integration of back-end data with external parties. Hours logged in the cloud stays in the cloud, and additional analytics cannot be performed on the data generated by the users.
Proofhub <sup>8</sup>	Project management platform for management and delegation.	The tool provides progress feedback and planning capability through the use of a Gantt chart. Online availability with categorised tasks for people to work together in groups. Chat functionality.	Cannot integrate with the existing platform since data is not available through the provided programming interface. Private company data is stored at a third party introducing risk.
Basecamp [9]	Collaboration software that supports file sharing. The tool keeps discussions and documentation in one place.	External API available for integration. The tool focuses on team communication and discussions.	Limited upload space for large companies. Encryption is only available for better-paid solutions. Teams prefer email over this solution.

<sup>7</sup> <https://asana.com/><sup>8</sup> <https://www.proofhub.com/>

Task Tool	Description	Advantages	Disadvantages
Todoist <sup>9</sup>	Full-featured, web-based, software platform. Built to allow collaboration between users. This is a task-based tool with a mobile application.	Easy to use interface. Email, task categorisation and user grouping functionality. Push notifications to allow progress updates and task planning capability.	A limited API exists to interact with the system, not analysing the data. No efficiency metrics are available. Lack of interface customisation and branding.

Table 1.1 shows the advantages and disadvantages for each task tool. A recurring pattern in the disadvantages column is that integration with an existing platform may prove difficult. The tool may not provide full integration, or the application interface may prove to be insufficient. Integration within an external tool is of high priority, due to the following :

- Integration will lower the risk of future application interface changes. MC van der Bank suggests that ever changing dependencies in a web application will negatively affect maintenance [44].
- Integration will lead to the business owning the data and minimise the risk of third-party data breaches, or misuse of the data. A business may suffer in various areas if the business data is compromised [45].
- The business will have to rely on the specific tool when an external tool is used. This will lock the available features to the subset of the tool and minimal effort expansion is no longer an option.
- The company may struggle to financially prove a reason to use the tool since most of the tools in Table 1.1 only have all their features available for a paid version. When the tool is integrated, a platform may have business value.

A conclusion can be made that integration is a necessity.

## 1.5 Problem statement

A specific project-based company exists with an existing web platform. This company requires a tool to manage their projects efficiently. Table 1.1 suggests that an easily integrable task management tool does not exist for the specific company. The existing applications and

---

<sup>9</sup> <https://en.todoist.com/>

their interfaces do not facilitate the needs of a comprehensive integrated program. It may prove difficult to manipulate and query the data generated from the tasks if a third party web interface (or API as mentioned in the previous section) is used from the previously-mentioned examples.

An external existing tool introduces inconveniences and time-consuming activities to an already complex system such as :

- The workflow of the company becomes implicitly dependent on the tool.
- Maintainability becomes a long-term problem, especially regarding interface changes.
- User accounts need to be created for the same people in the same company on different platforms.
- If the company needs to provide feedback on the projects, it would have to be manually pulled through a third-party system.

## 1.6 Need for the study

A need exists for a task-tool to manage the efficiency of projects. The task tool needs to be integrated with a custom interface with additional custom features. The user interface is important for branding and trademark, because inconsistencies in branding can lead to a loss of market share [46].

All the existing solutions contain arguably the same set of features, but are not fully comprehensive. A tool with the same core functionality that has multiple applications needs to be developed. Such a system will have a high functional value and can be resold and adapted for various applications. A need exists for an integrated platform that provides a foundation for a task management tool. This tool needs to:

1. Allow communication between users. A central hub per task must be available where all users have the same information regarding a task (Section 1.3.1 & 1.3.2).
2. Enable project planning. Through breaking down a project into tasks with deadlines, a broad overview is generated before project implementation (Section 1.3.2).
3. Keep a record of possible knowledge transfer. The tool must keep track of the communication between users and the objectives of the project (Section 1.3.4 & 1.3.5).
4. Integrate with the existing web platform. The task tool must use existing user accounts and be written in the same code base as the existing platform (Section 1.4.2).

## 1.7 Research process and implementation

The research process identified the problem of a task tool that is needed to improve project efficiency. To address the problem, literature that reflects the context of the study needs to be studied. Below follows the objectives of how the rest of the study will be developed.

### 1.7.1 Main objective

The main objective is to create an integrated task tool to keep record, enable planning and allow communication among users. Users can create tasks and assign other users to inform them about the work. A central list of tasks will give feedback of the work assigned to each other. Selecting a task will give an informative view on the detail and comments associated with the task. After tool development, it must be evaluated to see if it addresses the problem. This needs to be specified earlier.

### 1.7.2 Literature objectives

The aim of this study is to determine what features influences the efficiency of the processes in a company. The study aims to identify how knowledge transfer takes place and what the role is of human resource management. This includes project planning and record keeping. The study also investigates modern web-technologies and workflows to assist a practical implementation. From the need and problem statement, certain research questions will be answered in Chapter 2:

1. How does time efficiency and the management of a project influence the success?
2. How will the use of a task tool help improve the current situation?
3. How does a task tool help manage the human resources in a company?
4. What is record keeping and how does it influence knowledge transfer?
5. How does knowledge transfer take place?
6. What programming languages and web frameworks exist for a web-based application?
7. Which applicable software development workflow models are currently used?
8. Which web-based design patterns are applicable for the architecture of the tool?

## 1.8 Chapter overview

Below follows a short description of the content of each chapter. Each chapter has a specified goal and represents a part of the study procedure.

### **1.8.1 Chapter 1**

The chapter starts off by giving a short description of the efficient management of projects. Areas which influence the efficient management of projects are identified and then followed by a short background on task tools. The problem statement and need for the study is created from the information initially provided.

### **1.8.2 Chapter 2**

After problem identification, a study commences of the areas that define the efficiency of a project. Time efficiency and project management are researched to determine how to improve the efficiency of projects. After this has been established, a brief description is given of how human resource development forms a part of this study. A deeper look is then taken into record keeping to see how a task tool warrants knowledge transfer. Lastly, the software development processes, program architecture and programming languages are evaluated to see how the tool must be developed.

### **1.8.3 Chapter 3**

The design and implementation of the task tool are described in this chapter. The requirements are set for the tool, followed by the specifications and the outline of the system. The procedure that was followed during the design process is discussed. This includes both the front-end and back-end parts of the system as well as the implementation thereof. Lastly, the final phases that forms part of the chosen software development process, such as testing and user training, are discussed.

### **1.8.4 Chapter 4**

The results relating to case studies and a survey are discussed in this chapter. The set of case studies includes the use of the tool and how the task tool functionally meets requirements. The tool is validated through a predefined survey to determine whether it achieves the desired design goals. Certain statistics and findings that address the outcomes of the study are also presented.

### **1.8.5 Chapter 5**

The task tool is critically evaluated and a conclusion is drawn from the study. Further recommendations are provided for any further studies.

## CHAPTER 2

---

# Background of efficient management and software solutions

---

*'A man who dares to waste one hour of time has not discovered the value of life.'* - Charles Darwin

## 2.1 Preamble

In the previous chapter key areas were identified to improve project efficiency. These need to be explored further in this chapter. Firstly the management efficiency, time efficiency and project management traits are investigated. Thereafter the human aspect of a project describes how knowledge transfer uses record keeping to improve the efficiency of projects. The aforementioned concerns are addressed by following the technical details needed for the implementation of Chapter 3.

## 2.2 Time efficiency and project management

L. A. Ika investigates the factors for project success that can be attributed from PM journals [2]. The study describes the project processes and what defines project success. Planning and communication lead to better use of the project resources, as well as time efficiency. The following section describes how time efficiency and project management influence the efficient management of projects.

### Background on time efficiency and project management

‘Application of knowledge, skills, tools, and techniques to project activities to meet the project requirements’ defines project management [47]. In the world of today, technology can be used to apply project management techniques with the intent to be more profitable and efficient [48]. Contrarily, this does not mean that the processes of a company will be more efficient through the use of the latest tools and technology.

The organisation has to use the tools, and use the tools correctly, to gain an advantage in time efficiency. Time efficiency is coupled to project management within a business organisation [49]. Project management is in itself an activity which relies on predetermined principles or phases to be applied, such as planning, implementation, verification and various other stages of a project [5, 10, 18]. These phases, or principles, differ for each project and require adaptability [13].

Adaptability is the ability to respond to change in the environment. The environment of a project changes and progresses as the project evolves, and might damage the efficiency if not handled correctly [36]. Project efficiency is defined as ‘The firm’s ability to generate/select and implement/execute projects skillfully’ and is ‘considered as one key metrics of project success’ [50, 6, 32].

‘Project efficiency’ refers to the success rate of the projects relying on cost, time and quality in other literary contexts [51]. The efficiency refers to an improvement or some value added to the project process for the study. Literature determines early some characteristics of the project (schedule, cost and agreed specifications) that affect the efficiency of a project negatively [3].

There is no silver bullet for determining all of these traits of the project beforehand. These traits are multi-faceted, and the project planning, integration and the success of the business influence the path a project will follow [52]. It may create a feeling that anything might go wrong for a business to tackle new ventures, and to avoid project failure, one has to avoid new projects.

It is, however, not the case. The idea is to be wary of the pitfalls in the project process and address them individually and situationally. Many issues decrease project efficiency and are blamed on internal and external factors [53]. Internal factors include inadequate supervision, scope creep or change in project direction and even an inexperienced team [54].

These are factors over which the company has some form of control. External factors include risks that the company may have no control over. Proper planning can, however, minimise the impact of external risk.

The project management of a company represents its competency and allows for a competitive edge [55]. Sub-categories, such as process theory, has emerged to describe the order of activities, as well as the duration of the lifetime of a project [30]. This field of study aims to allow proper planning and prevent situations, that consume more time and unfold during the project.

Guidelines exist to provide a framework on how projects should be tackled to help the planning process, such as ‘Project Management Body of Knowledge’ or ‘Prince 2’ [55]. These manifests are not yet all-inclusive since support for electronic forms of project management is not adequately described [49]. They contain themes, processes and risk assessment procedures to identify and mitigate problem areas within an organisation.

These activities are important during the project life cycle [24]. More and creative solutions are contributed by employees if their work environment suits them and they feel comfortable. If the project management process is handled correctly, more work can be expected from

employees and thus prove to be more sufficient. Aid on the complexities mentioned above can be provided with the use of the software platform.

### **Practical implementations of project management**

C. Desmond describes a subset of tools that are used in project management practice today [56, 57]. The articles indicate that a firm or team uses different tools according to their need. A scenario would describe the team creating a work breakdown structure (WBS). This would enable the team to facilitate unforeseen circumstances and do financial planning.

The project management tools would allow the following during the implementation of the project:

- Team collaboration.
- Planning and cost estimation.
- Displaying all relevant information regarding the project.

A risk assessment is also necessary [12, 54, 55]. Risk assessment is done by identifying the risks before and during the project process, and keeping track of who will carry this risk in a risk register. Tools help identify the risk by their impact, probability and cost to the company.

### **Relevance of project management to the study**

The proposed task tool is aimed at assisting the time efficiency and project management process through simplification and planning to address the internal and external risks. Karkoulou et al. describe a work environment with multiple levels of employment and the importance of communication between the different skill levels in the organisation [58]. A distinction in the employment hierarchy is between the manager and the employee. Communication allows a better implementation of the final product [58]. The software platform will provide an integrated solution that delivers a communication mechanism and allows risk identification through planning.

Browning and R. Tyson describe complex project processes and the corresponding activities [59]. From the complex processes it is clear that project activities need a definitive description and status. The project management techniques should also describe how long a task should take and the order in which activities should take place. A task tool will enable project activity supervision to promote a productive environment.

### 2.2.1 Human resource management and development

A company might find it difficult to exist without employees. For a company to be successful in the long term, it has to develop and invest in its employees [5]. Human resource management (HRM) is a vast area that covers multiple practices [60].

A. Pak et al. describe how human resource development (HRD) takes place in business teams [5]. The article describes how organisations send their employees for educational programmes at the expense of the firm. These endeavours sometimes do not achieve the desired goal. The article suggests that a structured framework for HRD is needed to get the most out of the employees.

This framework comprises a set of variable processes, mechanisms and factors in the firm such as [5, 53]:

- Clear communication
- Alignment of project and strategic goals
- Successful allocation of resources
- Well-defined project requirements in the WBS
- Proper schedules in the project management plan (PMP)
- Feedback on the completion status

These factors, addressed by the project framework, will likely lead to a more successful outcome. The software platform developed for this paper aims to provide a mechanism that enables human resource management as well as development. HRM is done through work delegation and the provision of a planning platform. A task-based tool splits and assigns work to several employees, as mentioned in Chapter 1.

The mechanism that enables HRD for employees as well as the respective talent, is done through record keeping and knowledge transfer. A task-based tool logs all the work assignments and the conversations that took place regarding the respective topics. Through a well-oiled project management system in the company, human resources can be managed efficiently. Efficient management leads to the probable success of projects that the internal employees are involved in.

### 2.2.2 Record keeping

Record keeping refers to the documentation of events and procedures during the project implementation phase [13]. Project record keeping provides a mechanism for knowledge transfer [33].

In a typical scenario the manager or employee logs the responsibility of each worker, which creates a sense of ownership that influences the quality of the work [61]. Each person working in a group receives delegated responsibility.

A means is created to quantify and measure the project performance through logging the work. The records generated during the process can be used to evaluate whether the performance was satisfactory after the project implementation. When progress measurement can be applied, the course of the project can be adapted to maximise efficiency.

Work contracts may require a specific number of man-hours. The company evaluates the implementation of the project requirements with the use of a proper record. In another sense, from the viewpoint of the stakeholders, the documentation can be valued with the Return On Investment (ROI) [28]. Documentation may prove invaluable for a project.

The software platform solves this problem by making documentation part of the project process. There are still questions in the literature that needs to assess exactly what knowledge is appropriate or critical for each project [28]. The first step is then to inject an assessment phase into the knowledge transfer process. The software platform and task tool aim to make record keeping and documentation part of the project process through the use of the tool.

### 2.2.3 Knowledge transfer

A firm discovers only certain aspects that challenge the success during the project process. These come forth in the procedures and mechanisms followed during the flow of the project. To benefit in the long term, a company needs to retain previous solutions to the knowledge. After knowledge retention, a means should exist for the employee to learn the new knowledge.

The company has to ensure that the processes and tools are available to enable learning. These can be in the form of recording the actual events of the project and assessing the lesson impact. The origin of new knowledge also comes from the creativity of the group in which solutions are produced to solve project complications. If there is a lack of incentive, or the

group does not commit to participating in solving these problems actively, new knowledge may not stem inside the organisation at all [35]. For the process of knowledge transfer to take place, the following has to hold [35, 33]:

1. The recognition of the value of the knowledge.
2. The willingness to share or transfer the knowledge.
3. The medium in which the knowledge is transferred.
4. The willingness to learn the new knowledge.
5. The ability of the recipient to learn.

A company in a real-world application should have the following procedures in place to ensure that knowledge transfer takes place [13, 15]:

- Keep track of tasks in the firm.
- Build or use tools that have embedded knowledge.
- Document procedures and rules.

The task management tool provides a platform for record keeping and enables the business platform to log any insights gained from previous experiences. The only way that knowledge transfer will be able to take place is if the tool is actively used for this purpose and if the working environment supports learning during the project phase. The tool therefore addresses the first and third points of the knowledge transfer process mentioned above.

### **2.2.4 Project planning**

Literature refers to several planning methods. These methods contain goals which include team development and project- and behaviour management techniques [62]. The planning of a project can also include human resource development.

During the implementation phase of the project, this plan is used as a reference when decisions are made regarding the flow of the project. Clear communication is needed from all parties in the project to ensure that these decisions flows according to the project plan [63].

J Emblemståg describes how planning takes place in a firm [14]. Future planning requires some relative information from the past. This information is used to project to the future. Planning is not only done for possible events, but also for coordination. This entitles ‘communication and the elicitation of tacit knowledge’ [14]. Record keeping would benefit in such a situation.

The suggested software platform for this paper requires a clear communication platform among users. When a project is assessed before implementation, especially in the IT sector, it has to be broken down into some form of functional specification. These specifications serve as parts to be completed for the system during the implementation phase. Many development models, discussed later, thrive in such an environment.

The platform developed in this study forces the user to break complex projects down into practical steps. Each step is then assigned to a user with a possible due date. The steps are thus enforced and an outline of the project milestones creates a plan.

## 2.3 Software development workflow models

Section 2.2 described the theoretical implications of a task-based tool in the firm. The forthcoming sections review literature that describes the tools, design patterns and architectures that will be needed to implement the software platform.

A certain approach has to be taken to develop the software platform. An engineering approach has to be taken to assure the best software quality for the platform. Below, in Table 2.1, follows a list of existing software development models of which only the most applicable will be discussed further [64] :

Table 2.1: Types of software development workflows

<b>Traditional</b>	<b>Contemporary</b>
Waterfall	Agile
Incremental	Extreme programming
Spiral	
V-Shaped	
RAD	

### 2.3.1 Waterfall model

The first and most simple model gets its name from its linear downward nature. It is often not the best choice for software development, due to critique received in literature [26].

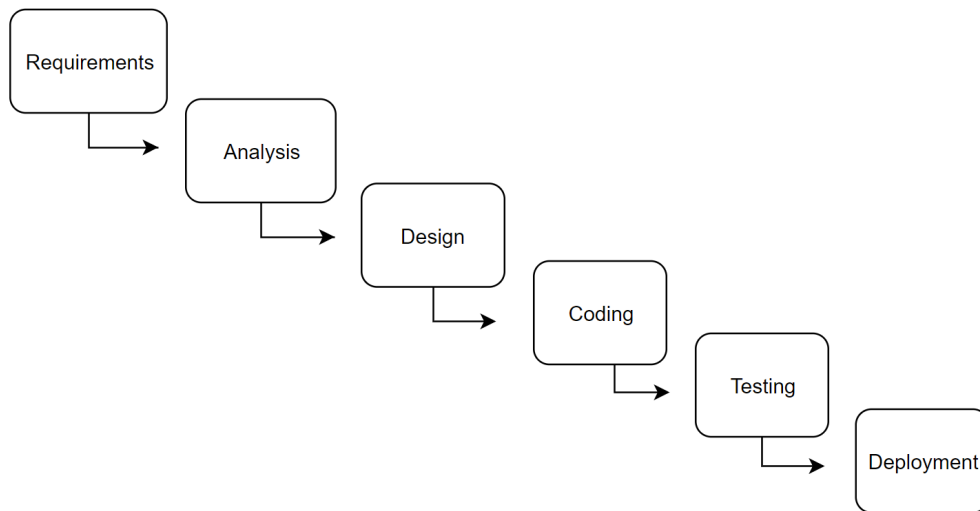


Figure 2.1: Functional diagram of the waterfall model

A description of each block in Figure 2.1 follows:

1. Requirements are set for the project and compiled into a software specification.
2. The framework and architecture of the program are designed.
3. The project is implemented. Actual workers come in and build the software according to the specification.
4. The functional parts are integrated and tested after code completion. This contains both unit testing and integration testing.
5. Continuous maintenance on the code is done after project delivery.

The waterfall model does not adapt well to changing requirements due to the entire process having to restart. The waterfall model may still be used today if a rigorous software implementation process needs to be followed.

### 2.3.2 Iterative and incremental method

A visual description of how an iterative software development method is implemented is given in Figure 2.2:

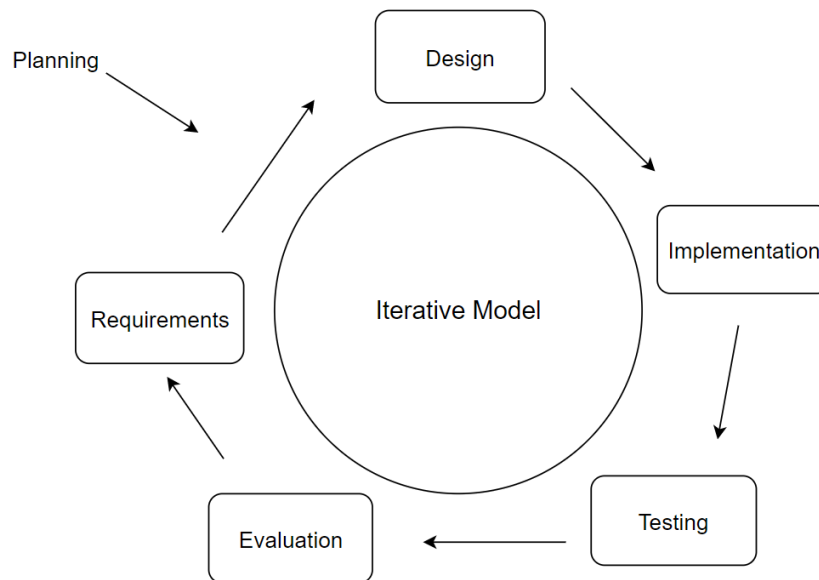


Figure 2.2: Functional diagram of the iterative model

The incremental model describes a subset of smaller one-way models [27]. A set of requirements is broken into functional parts. Each functional part treats its own goal or milestone and runs ‘independently’. The task software, for example, needs a feed to show all the active tasks and a way to view the task itself. The two requirements may be implemented separately and concurrently with a smaller approved model, such as the waterfall model. This process happens iteratively.

Continuous improvement on existing features is applied. Addition of new functionality adapts the current version of the software with each iteration. What differs in this approach from others, is that the entire software product is not readily available after a single iteration or function has been implemented. It is only released into production after the entire solution has been completed.

### 2.3.3 RAD and prototyping

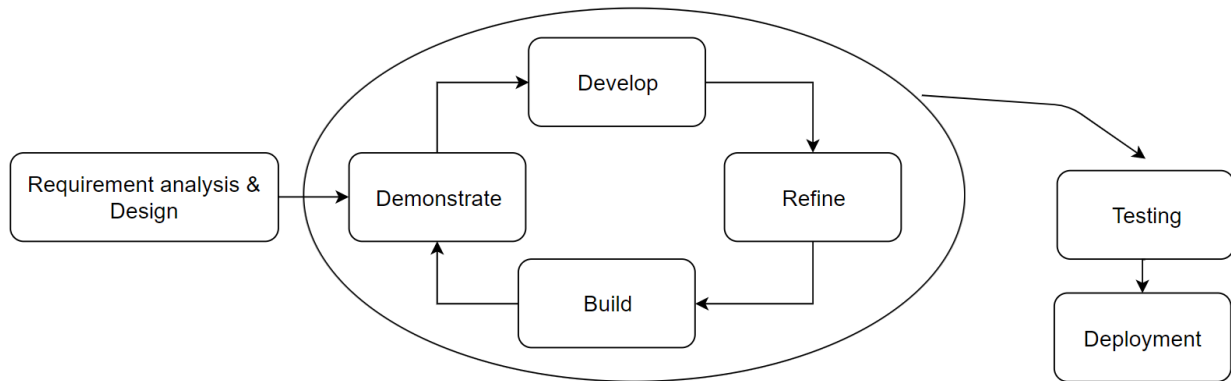


Figure 2.3: Functional diagram of the prototyping development model

The development phases of rapid application development (RAD) is shown in Figure 2.3. What sets prototyping apart from the other development methods is that, at the end of each iteration, a semi-featured product is delivered. Each iteration has certain functionalities of the final solution, alongside an actual implementation for demonstration. Adaptability leads to shorter implementation times and from this stems a term called ‘software tailoring’ [65]. If employers wish to shape and plan the business idea and the development situation requires high adaptability, this software development model tailors to the situation. Applying this model enables rapid adaptability, along with the user envisioning a physical representation of the software.

On paper the model looks attractive, but issues arise in practice. Integration and testing waste time as well as the additional development of features or functionality. With each iteration, a full set of testing has to be done on some features that are not even needed in the end product.

Notice that RAD and prototyping are different processes. RAD may lead to prototyping. With prototyping, the first product is rarely the final. A final product may be produced during the first iteration through the correct implementation of RAD.

### 2.3.4 Agile development

Agile development does not only focus on the development of the solution, but also on how the work is delegated by taking the flaws of the previous methods into account. Agile suits a changing or evolving environment with set functional requirements [28]. The model boasts an incremental, co-operative and adaptive workflow. Agile is best suited for small teams

implementing and finishing the requirements of the software.

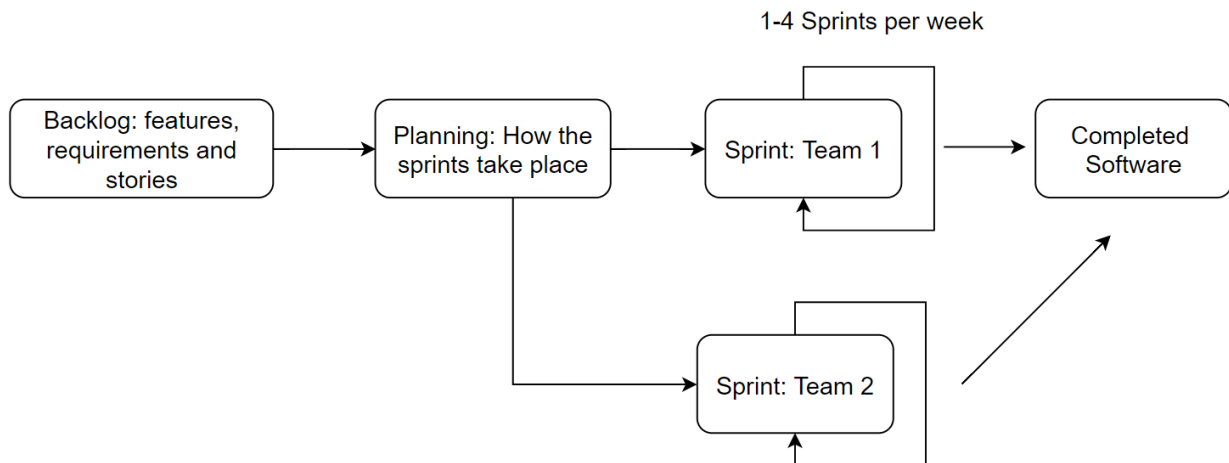


Figure 2.4: Functional diagram of the agile development model

Figure 2.4, above, shows how an agile approach tackles a new project. The software solution is functionally separated and put into a backlog. Through planning the features start being implemented in what is called sprints.

A sprint is a set amount of time in which the work needs to be completed. This allows some leniency in the time required for implementation. It also provides an incremental building solution to the product. This model allows workflow scaling to allow more people to be put on the project.

## 2.4 Programming languages for web-development

A diverse set of solutions exist in computer science and in practice to solve a web-based platform problem. There is a vast amount of programming languages available, each with its respective framework.

The software platform for this paper is developed on an existing platform for integration. Many alternative popular languages exist, each with its benefits [66]. Below follows a summary of some alternative web technologies:

Table 2.2: Comparison of programming languages with respect to web development

Language	Advantages	Disadvantages
JavaScript, TypeScript, Nodejs <sup>1</sup> , Express [67]	Very high community support. Client- and server side is written in the same language. Rapid development. Single threaded, asynchronous operations.	Still evolving and rapidly changing APIs and dependencies. Language specification changes every few years causing, faster ageing of code.
C# and ASP.NET with MVC <sup>2</sup>	It is a type-based language with community support. Security is taken into consideration. Diverse applications. Many services and integration APIs available for scaling.	Rapid development becomes complex to manage. Takes experience to apply the production-ready code. This language requires tedious integration with other languages. Older versions of the language run only on Microsoft machines.
Python and Django <sup>3</sup>	Scripting language with rapid development. Many independent packages that provide data analysis. Highly sophisticated web server.	Knowledge of full system is required to work. Monolithic system design. Spread across multiple implementations from different parties.
PHP and Laravel <sup>4</sup>	Fully Object Orientated and Dynamic. Cross-platform; Runs on Linux and Microsoft. Very fast to learn and develop.	'Relatively' secure. Readability, maintainability and performance do not necessarily meet with competitor languages.
Ruby on Rails <sup>5</sup>	Swift development. Many packages are available for various implementations. The ability for testing and automation during production shines.	Relatively old language. Documentation is sometimes difficult to understand. Threading and speed are 'slower' compared to other languages. Readability and maintainability of the language prove to be difficult.

<sup>1</sup> <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-javascript-full-stack-development/>

<sup>2</sup> <https://krishnamraju.jimdo.com/c-sharp/advantages-and-disadvantages/>

<sup>3</sup> <https://hackernoon.com/advantages-and-disadvantages-of-django-499b1e20a2c5>

<sup>4</sup> <https://www.quora.com/What-are-the-Pros-and-Cons-of-Laravel>

<sup>5</sup> <https://www.madetech.com/blog/pros-and-cons-of-ruby-on-rails>

Language	Advantages	Disadvantages
Java <sup>6</sup>	Platform independent. Widely used worldwide due to platform independence. The language provides enterprise-scale solutions for web-problems.	Verbose language. High learning curve. Language inherently 'slower' as compared to C #. Language previously experienced security issues causing the language to lose some popularity.

The requirements of the existing web server dictates the language. These are listed below:

1. Speed is needed for back-end processing specifically. The web service itself does not need to deliver web pages as fast as possible due to web-technologies having great variance in delays for providing web pages to the client. The system does, however, require multiple back-end processing with a variety of data sources. A general decision was made to create one code base or web platform to deliver pages, as well as to enable calculations. Speed would therefore be a necessity.
2. Security is an essential part of the language and framework. Due to a large part of the code base being exposed to the outside internet, security would be a concern. An existing framework, designed with security in mind, would ensure peace of mind.
3. Support for the language itself and the framework is needed. A diverse group of developers with different experience levels work on the code base. Good documentation would speed up development time and allow troubleshooting.

The main web application was chosen to be developed in C # within the ASP.NET environment. Cross-platform implementations would thus prove to be difficult, but the company mainly used Windows as a software platform.

## 2.5 Web-based design and program architecture

Interface design within web technologies come with a specific set of challenges. These challenges pose a threat to the development process of the platform, as presentation logic and processing only have a few viable, elegant solutions. Design patterns were created and proved to be the standard way of dealing with web-design problems to prevent re-inventing the wheel. A few key terms need to be defined first:

---

<sup>6</sup> <https://www.infoworld.com/article/2687995/java/4-reasons-to-stick-with-java.html>

- **Browser:** The web browser is a desktop application that has the inbuilt functionality to send and retrieve data over the internet and may be able to interpret HTML.
- **HTML:** Hypertext Markup Language is a universal markup that is supported by several web browsers and is used to render the elements on the page. Each element (e.g. button, input box) that is displayed on the web page is represented within tags with certain properties (data or visual).
- **View:** The view represents the HTML and JavaScript combined that is sent to the browser to draw a user interface. The purpose of a view is to contain as little processing logic as possible. The view renders content for the front-end.
- **Model:** The model represents a container for data. The goal of the model is to group and contain similar data that needs to be passed through the program.
- **Controller:** A controller is a separate piece of code that contains only logic and no HTML or JavaScript. The controller has the goal of accumulating and processing data for the view. The controller handles the back end.
- **ViewModel:** The ViewModel is another data container, but with the purpose of only providing data from a controller or presenter to the view. This may also contain additional data that only a certain view needs.
- **Presenter:** The presenter has the same functionality as the controller, but any additional changes in the model, has to happen through the presenter.

A few applicable patterns, which are considered to be widely used, are discussed:

### 2.5.1 Model View Controller

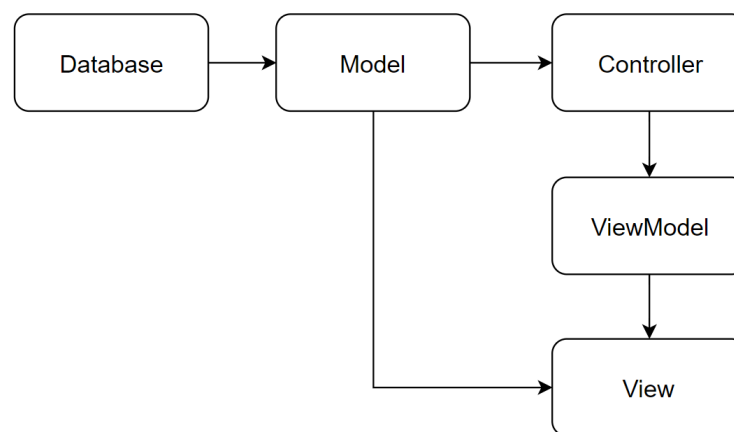


Figure 2.5: Components of the MVC pattern

Figure 2.5 shows the data flow through an MVC application. The controller gets data from the database in the form of a model. This model can be sent to other parts, libraries or

components of the program for further processing. An optional ViewModel may be generated to pass the processed data with additional aggregated information, such as the user ID number, to the view. The view processes the information and generates code for the browser to understand. The principle behind the design pattern is to keep the data, presentation and processing logic in separate areas of the code. This enables easier debugging and code quality. Another challenge the design pattern solves is program flow. The representation of the state that the user has to interact with only lies in the view. The controller or models do not have to keep track of the current state the user sees.

### 2.5.2 Model View Presenter

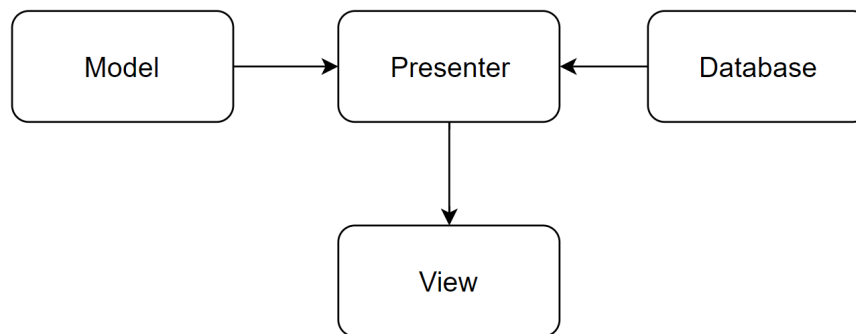


Figure 2.6: Components of the MVP pattern

The model view presenter is a derivation from the MVC pattern. There are only a few slight differences <sup>7</sup>:

1. The model does not know about the view and vice versa. It allows for easier debugging since changes to the presentation only happen in a certain portion of the code.
2. Each presenter may only map to a single view. This simplifies the logic, since multiple views may be returned in the MVC pattern.
3. Testing is inherently easier due to the simpler nature of the pattern, and all of the logic lies in the presenter.

### 2.5.3 Model View ViewModel

The MVVM architecture is more simplistic than the MVC pattern, with the view model containing all the logic and processing for the view <sup>8</sup>. The model is only a representation of

---

<sup>7</sup> <https://www.infragistics.com/community/blogs/b/todd-snyder/posts/mvc-or-mvp-pattern-whats-the-difference>

<sup>8</sup> <https://medium.com/flawless-app-stories/how-to-use-a-model-view-viewmodel-architecture-for-ios-46963c67be1b>

the data coming in from the database. The three blocks of the architecture are depicted in Figure 2.7. An interesting concept named data binding is used to enable data to flow from the view model to the view. The content of the view is created and updated automatically by updating the view model. What happens in the view model is thus kept in sync with the view and no other mechanisms have to be implemented to update the view.

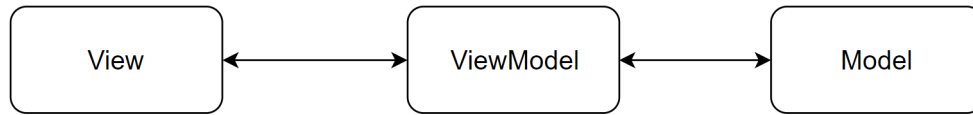


Figure 2.7: Components of the MVVM pattern

The characteristics that distinguish MVVM from the others include :

1. The view model not only contains the data for the view, but it also contains processing logic.
2. The view is updated through data binding.
3. The model only contains data from the database, but nothing restricts the model to have additional properties that represent data for only the view.

## 2.6 Literature objectives

Chapter 1 asked several literature questions that were implicitly answered through the previous sections. A summary of the explicit solution to the proposed objectives follows.

### **How do time efficiency and the management of a project influence the success?**

If project management is done well, the project has a likelihood for success and can reap more profit. This is not an easy accomplishment, due to several factors, sometimes outside the control of the team, that might influence the project outcome. Time efficiency in project management reduces the cost and several risks associated with the project. If the people are handled efficiently, the time consumption of a project has a higher probability to be lower.

### **How will the use of a task tool help improve the current situation?**

The task tool covers all major areas that efficient management needs:

- Human resource management: Provides a mechanism to allocate people to tasks.
- Process efficiency: Monitors and provides feedback on all ongoing tasks.
- Record keeping: Logs all meta-data linked to a task.
- Knowledge transfer: The user may use the records to identify and extract valuable

knowledge.

- Planning: Tasks create an outlying plan for when to do what.

### **How does a task tool help manage the human resources in a firm?**

A task tool allows employees to delegate work and inform other users through linking them to a task. The tool enables a manager to effectively delegate work and group the employees.

This allows communication and a ‘hub’ of information available to the linked users. The employees have several necessities of the project information at their disposal. The users now have a tool to help manage themselves.

### **What is record keeping and how does it influence knowledge transfer?**

Record keeping encompasses the preservation of knowledge in the business. The knowledge gained through the project process is valuable and becomes embedded inside the people and processes. Proper record keeping enables more efficient knowledge transfer, since the important information is collected and in writing.

### **How does knowledge transfer take place?**

Knowledge transfer takes place through several mechanisms as listed below:

- Formal education: A business or individual endeavours to implement a structured plan to learn a new skill.
- Through work experience: An individual gains the knowledge through learning on the job.
- In the processes and procedures of the company. The way things were previously done is preserved in a company.

### **What programming languages and web-frameworks exist for a web-based application?**

A wide variety of languages exist summarised in Section 2.4. This list is not extensive or all-comprehensive. It only contains a list of many current relevant languages. The existing web platform was created in the Microsoft ASP .NET framework with good reason. This is also explained in Section 2.4.

### **What applicable software development workflow models are currently used?**

Table 2.1 lists all the relevant software development workflow models. Section 2.3 discusses them in more detail.

## Which web-based design patterns are applicable for the architecture of the tool?

A shortlist of popular design patterns include:

- MVC
- MVP
- MVVM

These are discussed in more detail in Section 2.5. The ASP .NET framework uses the MVC design pattern.

## 2.7 Conclusion

A definite need exists to handle projects efficiently in all areas of the project process. All areas, from project management to human resource management, would aid from the tool. In this chapter, it can be seen that a task tool solves the various needs in different ways. Various web architectures and languages exist to create a web platform for a task tool. The design choices and a short discussion of what and why a specific software design process was chosen follows in the next chapter.

# CHAPTER 3

---

## Design, development and deployment of a software platform

---

*'Digital design is like painting, except the paint never dries.'* - Neville Brody

## 3.1 Preamble

From the literature discussed in Chapter 2, requirements are created in this chapter to describe the functionality of the task tool. A discussion of the design and implementation takes place. The system design is split into two pieces, namely the back end design and front end design. This is described in further detail in this chapter. The last part of the development process includes the testing and user training. The next chapter will discuss the results after implementation.

## 3.2 Processes and models

Considering the software development models discussed in the previous chapter, a combination of the iterative- and agile approach will be taken. The requirements of the system, especially during the implementation, changed. The agile approach suits such an environment very well. Prototyping and RAD involve several versions of a final product with different functionality. The task tool will have the *same* functionality from the start, only with additional features implemented after the initial release.

This is where the iterative model is also applicable. Each additional feature will be added iteratively. The solution is broken down functionally and each function will address a requirement. The functional components will be created through sprints. A selection of employees will test the software and additional improvements will be listed as the next set of requirements after release. This process is repeated until user satisfaction is met.

The task system is the main attraction of the solution. Figure 3.1 shows the systematic approach from development to deployment. The course of this chapter will follow the design process:

1. The system is initially designed with a set of requirements.
2. The system is released for testing after the core functionality is built.
3. From the testing, smaller incremental requirements are added and patched onto the system.
4. The tasks system is finally released to the company for use.

The results of the system will be verified and validated in the next chapter.

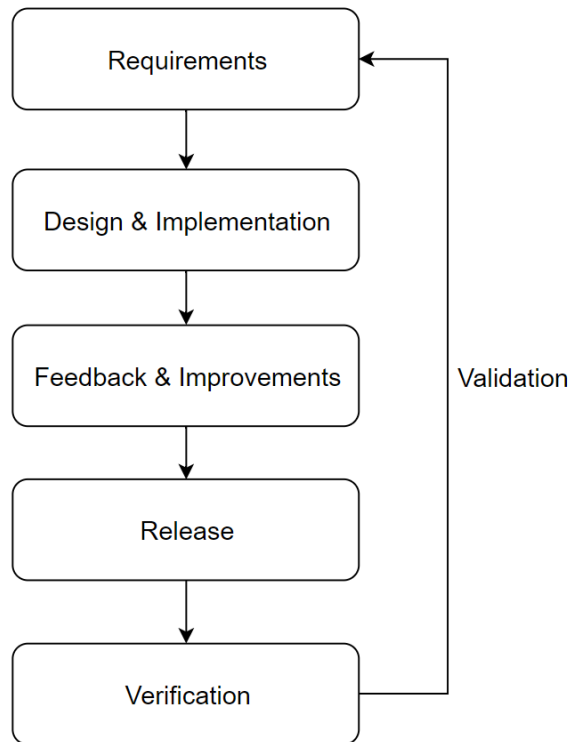


Figure 3.1: Steps followed in the creation of the software platform

### 3.3 Requirements

A problem needs to be well defined for it to be solved. The software is a tool or means to solve a problem. Any problem presents itself as a subset of characteristics that needs to be handled. Even if all the sub-problems are solved fully, the problem itself may only be solved partially. A well-defined set of requirements will prevent this situation. The requirements define how the problem needs to be solved and if the software will solve the problem <sup>1</sup>.

The requirements for this study will be broken down into:

1. Business requirements
2. User requirements
3. Functional requirements

#### 3.3.1 Business requirements

Proper planning does not start with code implementation, but rather with a basic idea formulation [68]. Business requirements create a broad overview of the purpose of the software and how it must be used to generate income. If no proper business plan sustains the moti-

---

<sup>1</sup> <http://rmblog.accompa.com/2012/04/types-of-software-requirements/>

vation for the project, the aggregate solution will be brought to life with time and money wasted. The requirements describe a way to reduce risk by defining the outcomes and save on development costs. The requirements also inform the stakeholders in the creation of the software what the expected outcomes are. The need and problem statement in Chapter 1 dictates a set business requirement for the software platform:

1. The system design must include a generic engine; the core must have more than one business application.
2. The solution is created to improve the efficiency of internal processes, such as projects through record keeping or knowledge transfer.
3. The task tool must be able to enforce planning through work delegation.

### 3.3.2 User requirements

User end requirements describe how the user will interact with the software and the needs which arise from this interaction. The users of the software include the entire company, both the employee and the employer. ‘360° feedback loop’ is a term which describes the way in which the internal sociology of a company requires all parties to listen and appreciate ideas [58]. Feedback is a process to reduce risk management.

Proper risk management leads to saving time and saving costs, thus improving the efficiency of a project [30]. The company that needs the task tool has specific groups of people who work together in teams. A project is allocated to each team. Individuals may also participate in multiple projects. Each team uses a different approach to complete their work requirements. This shifts the view of project management to something that emphasises the distinction of each process. These processes may contain the same project activities but are handled differently.

The user requirements were initially set through multiple feedback sessions describing processes from both employees and employers. The full set of user requirements is listed below:

1. Search and filter capability are needed to distinguish tasks from their metadata. This will include a way to filter according to priority and to search the tasks by name. The task metadata will be discussed fully in the next section.
2. The tool must be able to assign due dates for tasks to enable proper planning.
3. The users must be able to categorise and prioritise the tasks with some mechanism.
4. Certain users must be allowed only certain actions when it comes to a task, for example, if a person needs to only inform a user, he must not be able to generate comments.

The task tool must be able to restrict this action.

### 3.3.3 Functional requirements

This set of requirements describe what the system needs to do, and to show what functionality the implementation needs. This specification creates the technical system design. The system sees the task itself as a collection of information.

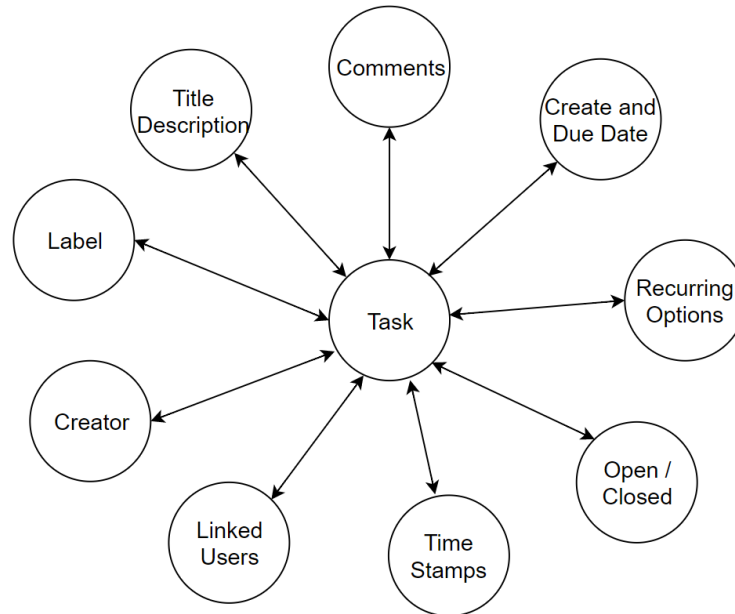


Figure 3.2: Task with different types of data that can be linked to it

This collection is depicted in Figure 3.2. Each part of the metadata represents a functionality:

1. **Title and description:** The task must have a title that must be short enough to be listed. The description is an optional longer text that provides a broader overview of the upcoming work.
2. **Comments:** Communication regarding a task must be allowed among users. If a user has further queries, or needs to provide verbal feedback on the topic, he may comment. This is a piece of text that links to the task with a time stamp.
3. **Due date and creation date:** Planning requires finished work in a set time frame. To delegate and keep track of the tasks, an optional due date is linked to the task. This enables prompt work delegation.
4. **Recurring options:** Some tasks may be of a repeating nature. The tool must have a mechanism that enables tasks to remind the user automatically of the appropriate information.
5. **Open or closed status:** The tasks must be able to be completed in order to organise

and keep track of how a project is progressing.

6. **Time stamp of each event:** Record keeping is an important part of the management process. Each change to the metadata must have a time stamp to record when it happened and how it was changed.
7. **Linked users:** Users may have different roles to play in a task. Different types of users must be linked to a task in order to gain access to all the information regarding the task.
8. **Creator:** The creator of the task must be logged on the system to keep a record of who initially produced the work.
9. **Linked labels:** Labels are a way to tag the data. The labels must prioritise or link status to the task.

## 3.4 Technical design

This section outlines the implementation of the solution. Each of the requirements from the previous section addresses a design feature.

### 3.4.1 Accessibility

Web-applications have many advantages over desktop software distributions. All clients have the latest version of the software and have access to it from wherever an internet connection is available. However, some disadvantages arise without an internet connection, as the application is seldom available offline. Flaws in distribution leads to all users experiencing the same bug. A web application is the best option for the infrastructure that existed in the initial application design.

### 3.4.2 Framework and system architecture

A motivation for the development of the tool above existing solutions is the possibility of integration. Two main integration opportunities are within easy reach given the current problem statement. An existing MVC web-based platform is already used to deliver services within the organisation. By using this platform, the project can reap the benefits of faster development by not re-implementing another web platform. Integration is furthermore possible with the back end software services. These include an e-mail and database service. The tasks will thus be tightly integrated with the entire existing system.

The framework for the task software platform comprises different physical software components, shown in Figure 3.3 below:

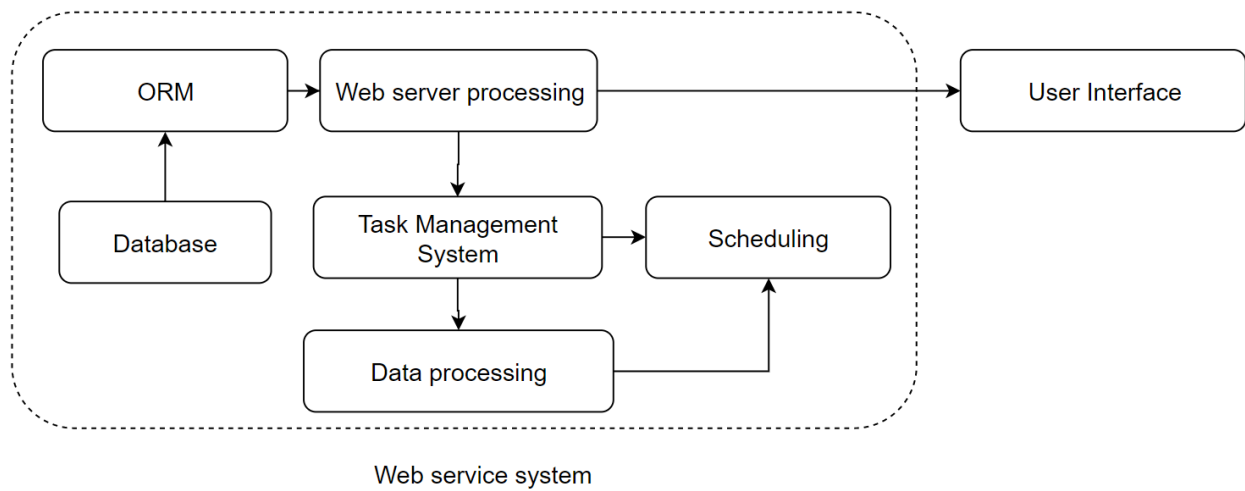


Figure 3.3: Functional framework block diagram

The entire process starts with the database design. This is implemented together with the Object Relational Mapping (ORM) functionality discussed in a later section. After that, the web server is configured with the appropriate functionality. This enables the aggregation of the data and delivering the web code to the client browser (user interface). After the configuration is complete, it is possible to create the task management system. The system is responsible for retrieving the data according to specified filters, generating relevant information for the user interface and communicating with a scheduling service. The detailed design and implementation of each block follow in the next sections.

### 3.4.3 Back end

The back end of the system is where all the processing and data manipulation takes place. This includes the following:

1. Database and Object Relationship Mapping (ORM)
2. Web server
3. Task management system and scheduling

The website's main function is the retrieval of data and then presenting it to the user. This is done in three phases, as described in Figure 3.4

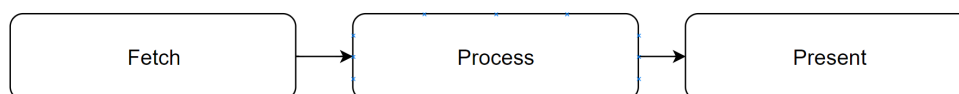


Figure 3.4: Back end functional flow

The fetch phase includes the database and ORM functionality. The process phase describes aggregation, grouping and presenting data for the front end user interface. The presenting phase describes how the server sends data to the user interface. The web processing takes place in the back end which is pre-built into the existing framework. This includes communication between the client and the server, generating data packets, and keeping track of the session. The framework allows data filtering, which is used for authentication and authorisation. Lastly, logical generation of client-side code is possible. Further detail on the operation of lower level technologies is beyond the scope of this study. The existing back end thus provides a full-featured set of tools that gives the creation process of the task tool a jump start.

### **Database and ORM**

The database that existed at the time of development is a MySQL relational database. It runs an InnoDB engine and uses the MySQL query language. Relational databases have the unique ability to try and prevent duplicate data entries. It also allows the modelling of data in such a way that it uses less storage space.

Many other relational databases exist that provide the same functionality as MySQL, such as PostgreSQL and SAP HANA. Another option for a database that became known in the previous decade is non-relational databases or NoSQL [69]. These databases provide a document storage approach. Instead of storing the data in tables with fields, all the applicable data is grouped in a document. The document itself can be found very easily within a large amount of data, but querying attributes within the document becomes an art.

The existing system also uses a form of a NoSQL database, but development for the MySQL database would be faster. ORM is a term used to describe the process where data is grouped from a source and mapped to objects that are used as groups of data and functions. Many tools, such as Entity Framework, exist in the industry, which makes ORM almost invisible to the developer. The existing system does the ORM by hand. The complexity of the data structures and objects arises from the system needs and makes it very difficult to use an external ORM.

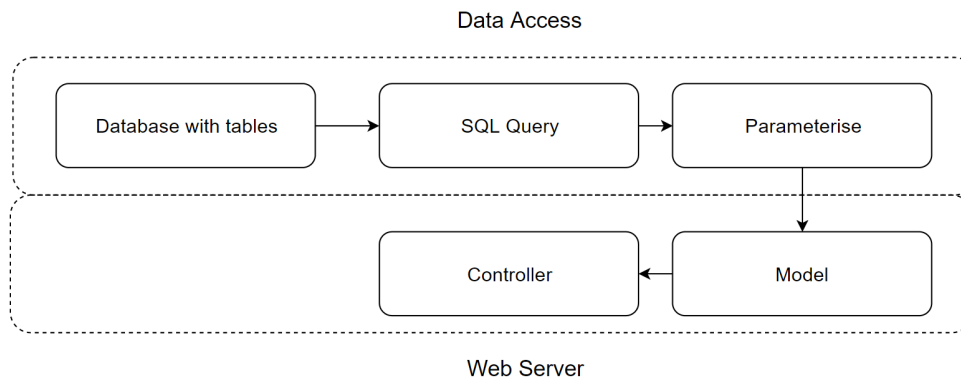


Figure 3.5: Object relational mapping

Figure 3.5 describes the process. The ORM starts off by creating a SQL query that is tested in the database. The query can contain any create, update, read or delete functionality. This is parameterised to ensure that SQL injection, for security reasons, is not possible. Next, a data model is created that truly represents the table. Upon request from the web server, the data is read into the models and provided to the controller. The controller is part of the MVC framework.

Significant time went into the back end record-keeping functionality. All the changes to the task metadata need to be logged. This implies that the back end had to detect changes in the data and log them accordingly. Initially, each time a change occurred within the metadata, the web service would write to the database. The logging of changes slowed down the system and created program flow complexities. The solution was to add database procedures. Each time a change to the metadata of the task happened, the database would internally write to the change tables to display in the task feed.

### Web server and MVC Framework

Model View Controller (MVC) is a design pattern that works very well with web platforms, as discussed in Chapter 2. This pattern can be implemented in any programming language that provides user feedback via an interface. The purpose of the programming interface is to fetch the task data and show it to the user via the web. The web application interfaces with the database, then generates pages for the user to view.

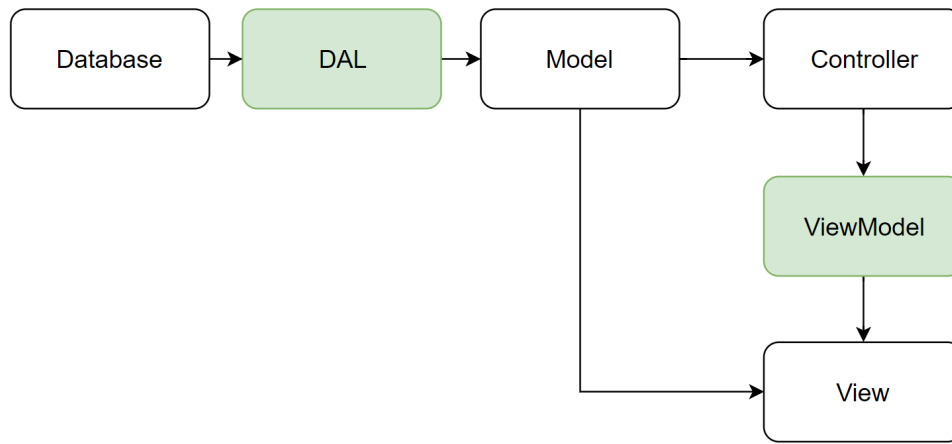


Figure 3.6: Adapted MVC pattern

A slightly adapted version of MVC exists in the current web application [44]. Figure 3.6 describes the operation. Two other optional components are added namely the Data Access Layer (DAL) and custom ViewModels. The DAL provides an interface for all database transactions and functions as the ORM. Custom view models introduce functionality for metadata for views that are not necessarily generated from the database models. A discrepancy between models and view models exists to ensure that empty data does not occur in the models that represent the data in the DB. The view models also separate view logic from data processing logic.

### Task management system, scheduling and data processing

The task management system is the functional component that is used to create, update, open and close tasks. Tasks cannot be removed or deleted for record-keeping purposes. Figure 3.7 shows how the task management system works to handle the data generated by the user:

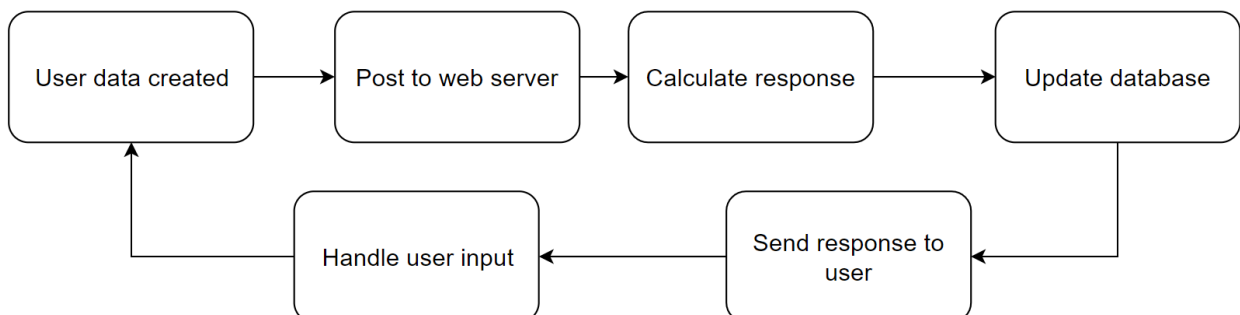


Figure 3.7: Functional system flow block diagram

Each time the user enters data and sends it to the server as an action, it calculates the

response and waits for the user interface or web browser to present new data. This data includes a wide range of comments and filter applications, or even functions such as assigning labels. The input generated by the user can either be data posted to the server to create/update a task, or be a request for a new page.

Figure 4.3 shows how data handling that is posted to the server takes place. Additionally, it shows how the response is calculated:

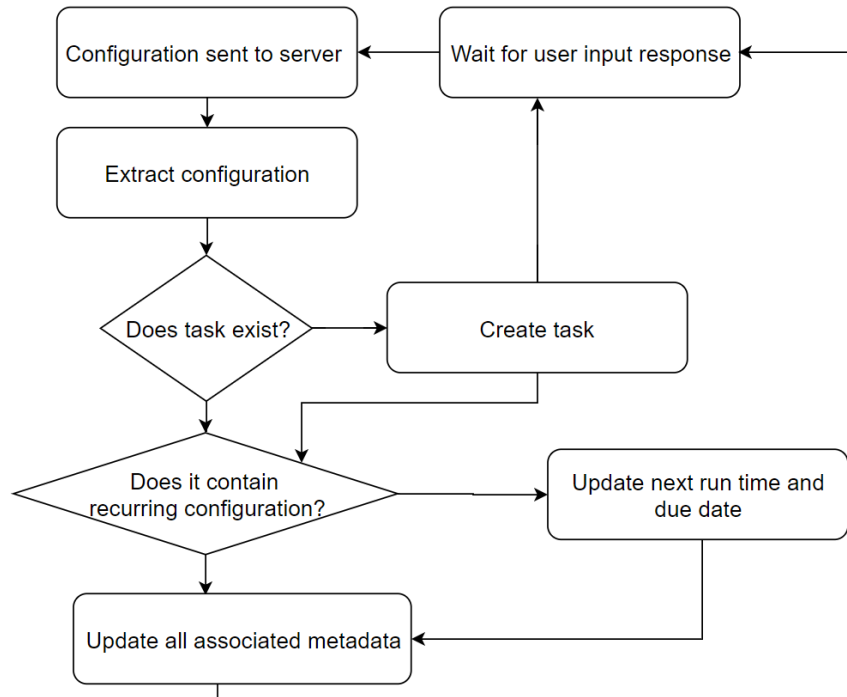


Figure 3.8: Create or update task block diagram

The process starts at the server, waiting for user input. The user sends the configuration to the server. The configuration is extracted, after which the server either creates a task or updates the existing task. The configuration is a JavaScript Object Notation (JSON) object. This object contains a list of key-value pairs with each key standardised from a predefined class and the values containing the variables. Figure 3.9 shows the JSON object.

```
aDayMonthWeek: "weekly"
aDays: "Monday"
aDueDate: "2018-03-04T07:00:00.307Z"
aFirstComment: "Short Description"
aFirstLast: ""
aFirstLastAmount: ""
aIsRecur: "1"
aMinutesDiff: "0"
aMonthOption: ""
▶aSelectedDays: []
aSpecialcaseSelect: ""
aTaskId: "0"
aTaskName: "Test Task"
aTimeDiff1: "3"
```

Figure 3.9: Example of a JSON key-value pair object

The JSON configuration may contain recurring options. The creation of such a configuration is a complex chore due to the nature of time calculations being particularly tedious. A later section discusses the full recurring configuration in more detail.

After the recurring options and the metadata have been calculated, the task is stored, and feedback is provided to the user. This feedback is sent back to the web browser in the form of a JSON object. If a user simply wants to request a new page from the server the diagram below describes how this takes place internally from the web service:

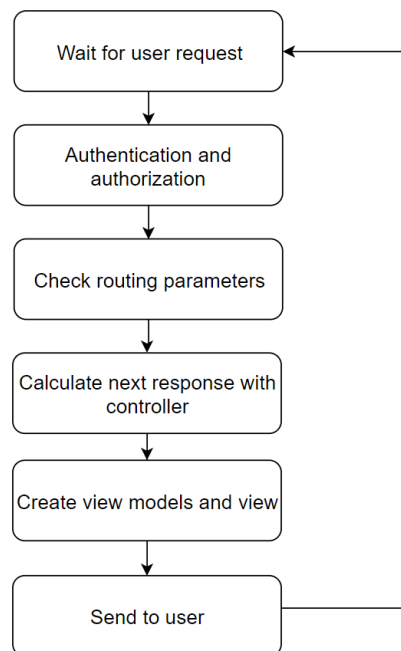


Figure 3.10: Routing block diagram

After the user requests for a new page, the entity is validated to check if access is allowed for the action. The server checks the requested URL, and the appropriate controller function is called to generate the view with the view models. Finally, the server sends HTML, CSS and

JavaScript to the browser [44].

Data processing primarily takes place on the server side. When the user desires to create a recurring task, the same algorithm is used in the front end to show the recurring dates for which the task will occur. These dates are calculated for the user to provide feedback on the possible options available during the complex recurring configuration. The process is done in such a way to ensure consistency between the functional system blocks. The data processing of the server describes how tasks will be filtered and queried for data analytics. Today, businesses treat data as a commodity [69].

Each time a user creates a task or a work log, the system generates data that provides valuable information. This information includes the time, number of employees and the project or concept worked on. Another component of the data processing allows the tasks to be grouped by their metadata (filtered) and ordered by linked dates. With the help of the database as well as several filtering processes, the system achieves a fast<sup>2</sup> aggregation of the metadata around the task.

When the user queries different types of data from the database, the complexity grows as the amount of aggregated data grows. Grouped data gets queried individually to solve the problem. Back end processing is needed to present it visible to the user to regroup this data.

### 3.4.4 Front end

The front end has two goals. The first is to receive input from the user and send it to the server. The other is to provide feedback from the server.

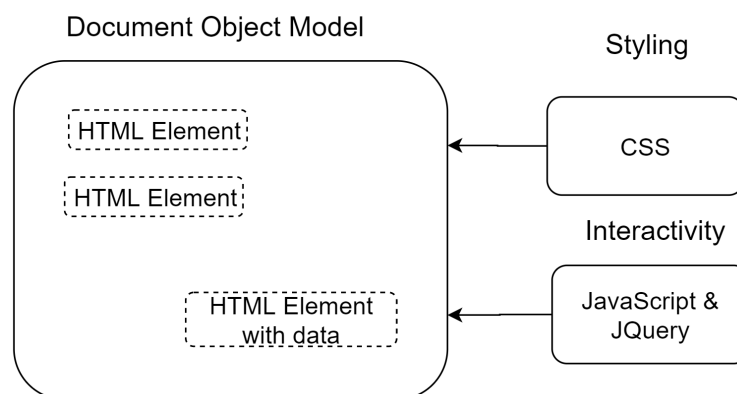


Figure 3.11: Front end browser block diagram

<sup>2</sup> Relatively speaking, fast in web terminology implies less than 1 second

Referring to Figure 3.11 the front end of the system contains the code that will run on the client side, or in the web browser. The web browser uses HTML and CSS to generate components and display the page. JavaScript and JQuery provide interactivity. Javascript is a scripting language that most web browsers understand and run in the background. JQuery is a collection of pre-defined JavaScript functions and is a popular JavaScript library provided available since 2006 <sup>3</sup>. It is a toolset within JavaScript that is very versatile and easy to learn.

The Document Object Model (DOM) describes the data-elements, such as divisions or buttons, that may be displayed from the browser. It allows the user to search for elements on the DOM and manipulate them fully. An action describes a submit button click. Each time such an action takes place on the DOM, either the interface is updated, or an action is sent to the server. The front end design is done through an iterative process.

User requirements dictate the design including the application flow and its functionality. Development-wise it is cheaper to redesign the front end than other parts of the system. Specifications are refined through the iterative process. The flow for the user interface is defined in Figure 3.12. Initial program flow allowed users to move among all pages at all times. The navigation flow was specifically designed to avoid user confusion.

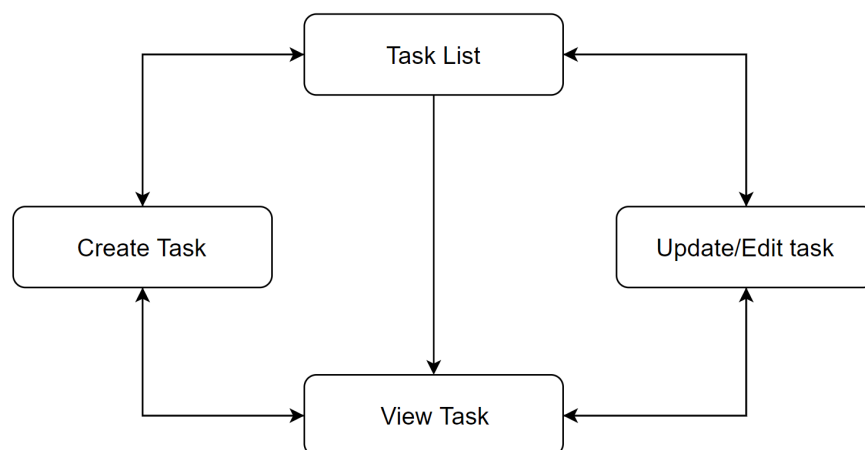


Figure 3.12: Front end flow block diagram

In essence the entire user interface can be described as a form coloured with animations and transitions. It guides the user to input the correct data that will be valuable to them when they are using the system. Each functional block represents a menu that is described in the

<sup>3</sup> <https://jquery.org/history/>

forthcoming sections.

### Task list

With reference to Section 3.3.3, a particular set of requirements in the design of the task list needs to be addressed. This design aims to address the functional requirements 1, 3, 4 and 6-10. The task list aims to display all relevant information of a task in one small view.

This information includes the title, description, due date, progress, linked users and linked labels. The ability to update and change data in this view is also a necessity. The existing web platform has a standard layout for the entire front end. This is depicted in Figure 3.13 below:

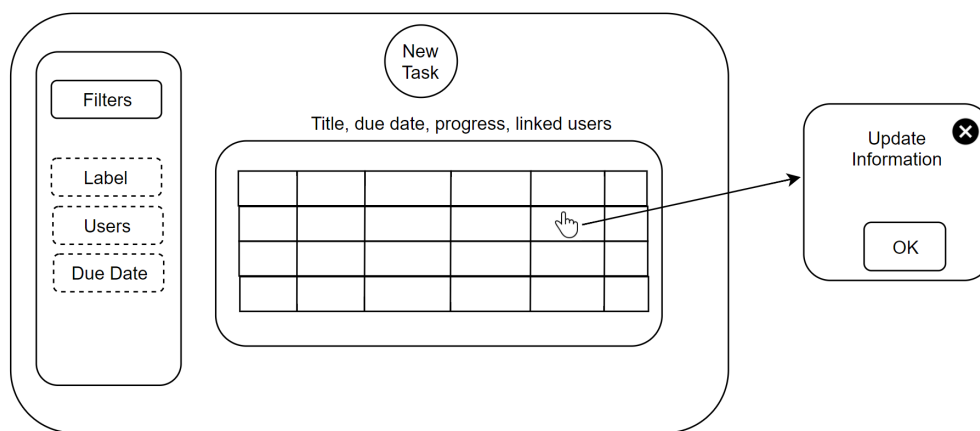


Figure 3.13: Task list design diagram

Each cell has its row and column. The row is the task selector. The column describes the linked information. The design allows this information to be changed by clicking on a cell.

A pop-up window with a save feature will send a new configuration to the server to update. Two important feedback tools exist, namely labels and task progress. The task labels tag the task with a description or a status. The progress of a task is a percentage indicating completion.

This differs from a label, in that a label might tag a task as 'busy', a one- or two-word description, whereas the progress gives an actual estimated value of completion. Filters for each type of metadata is available on the left side of the screen. The click event of a filter sends an action to the server which requests a new list of tasks. The task list thus serves as the central hub of information for all available tasks.

## Create task

This menu aims to meet requirements 1 and 3 of Section 3.3.2 and 1,3,5 and 9 of Section 3.3.3. The task creation menu was initially designed to not only create the task but assign any and all types of metadata associated with it. This proved to be too complex for the users since too much information was shown on a single screen. It was then decided to allow the user to only fill in the minimum amount of information to display on the task list. The task title is the only required field with an optional description, due date and recurring options. Figure 3.14 shows the initial design:

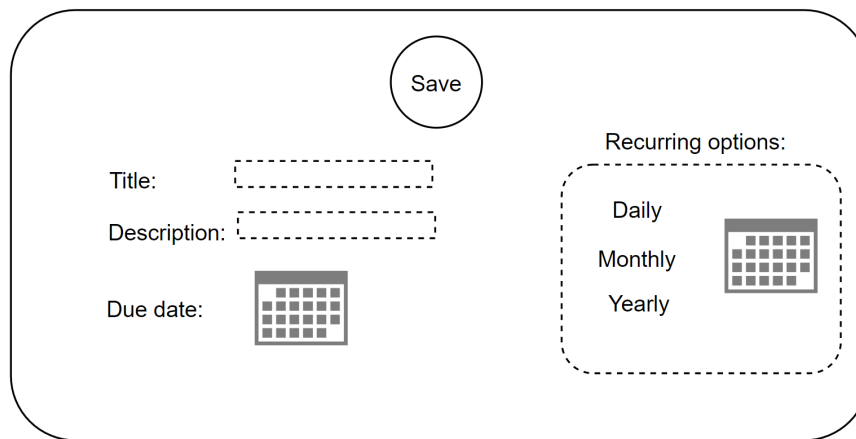


Figure 3.14: Create task design diagram

The due date can be added to a calendar picker with an intuitive ‘add or remove’ button. The recurring options, however, proved to be more difficult. For a task to be recurring, it has to have recurring options linked to it. The options are divided into time spans, namely daily, weekly, monthly and yearly. Each time span has a specific selection available.

The daily section has a time-of-day configuration for a new task. The weekly section has a day selection for each day of the week, and the monthly segment allows a selection for certain days of the month. A yearly section only caters for one day of the year.

When a user creates a task with recurring options, it serves as a template for the next tasks. These will be known as recurring triggers. When the scheduler uses a new task template, it supplies the data linked to the recurring trigger. The linked labels and users will receive a new task with the original description and title. The due date will be updated, using the applied configuration.

## View task

The view task menu provides a complete view of all the detail regarding the task. Almost all the metadata linked to the task is provided with the main focus on the comments. The view task menu aims to meet requirements 1,2 and 8-10 of Section 3.3.3 and 5 of Section 3.3.2. Figure 3.15 below shows the final design:

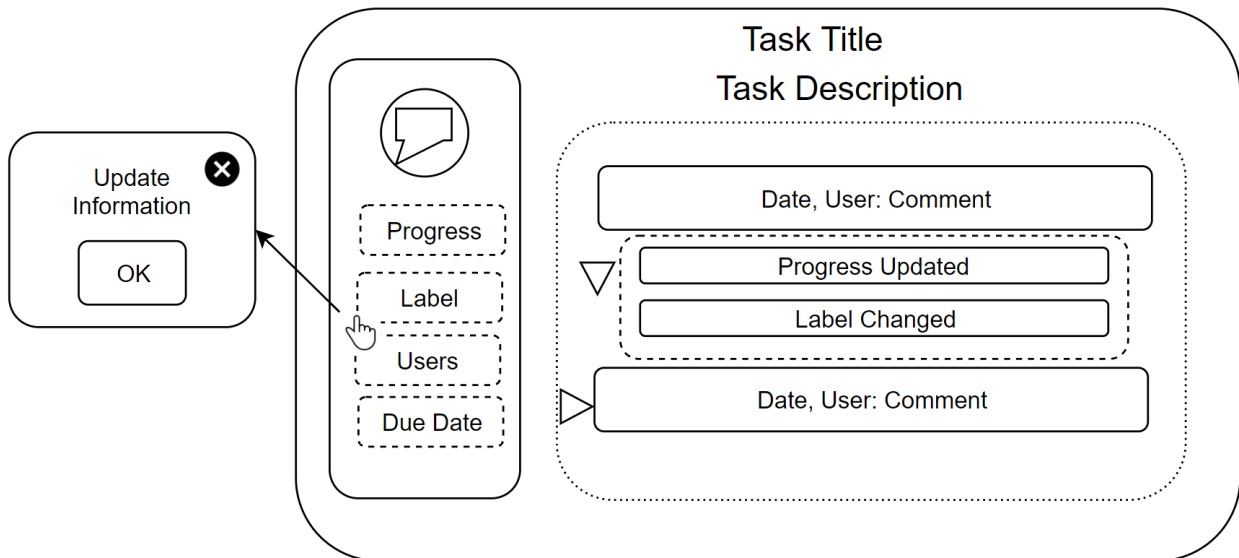


Figure 3.15: Task feed task design diagram

The initial design had a central task feed showing all relevant information, but due to the high frequency of changes in the task data, ‘noise’ occurred in the data. Only the comments are initially displayed with expandable blocks to display any changes that occurred. There will be a menu on the left side.

A new comment can be created by clicking on the ‘comment’ button. The comments also have the integrated email functionality of the web server available. Each time a user posts a comment to the feed, a popup dialogue enables the user to notify other users of the change.

Any change to the task would initially notify all the users linked to it. This feature was quickly altered due to a rush of emails being sent at task creation. The specific menu allows users to be assigned or informed or be a supervisor. An assigned user can open and close the task while the informed user only has the right to view the task. A supervisor accepts responsibility to follow up on the task.

The side menu allows the linking of labels, and the progress can be updated. The task options are displayed next to the due date. A button to the update menu is also available.

### Update or edit task

The edit task menu is the same page as the ‘task create’ menu. The same page is reused to give the user a familiar interface for configuration. The view task, as previously discussed, enables the user to change almost all the data that is linked to the task.

The due date, recurring options, title and description, are only available from the update menu. The previous configuration is loaded from the web server and displayed to the user. The server saves any changes and logs the changes that occurred.

## 3.5 Testing and user training

A system only reaches its optimum value once it is fully in use. The system can only accomplish this state if the user realises a need for the system and acquires the skills to use it. A user manual, which can be found in the appendix, increases the efficiency of the training process. The manual contains a step-by-step guide on how to use the features of the software platform. Each user receives a copy at initial release of the system. There is no guarantee that each user will work through the manual to gain the knowledge required to use the entire system. Consequently, the user interface is designed intuitively in order to guide the users to complete the actions they require.

Testing is a core part of the design process and may be done to evaluate the validation and verification mechanisms in the project. Testing on this platform is done during implementation and initial release. The latter guides the continuous development according to the chosen development model at the beginning of this chapter. The former is functional testing and input validation.

Unit testing is a software development practice that is used to test different code segments. Within the realm of the the web platform, tools exist to test user interfaces, but become time-consuming as the complexity rises. The risk value, coupled to an interface bug, is minimal, due to only a relatively small number of users initially testing the system. The actual users and not only the developers are dubbed as the testers of the system. System- or back end testing is done by the developer to ensure that the algorithms run as intended.

Both verification and validation will take place for this study to ensure that the software platform fulfils its goal. The business, user and several functional requirements are validated through the use of a survey (found in the appendix). A questionnaire with close-ended

questions is designed in the next chapter and it will target all the users of the platform. The outcome of this survey will determine whether the platform is sufficient.

The functional requirements are verified through functional unit tests. Each requirement is part of the design. A unit test would entail naming the specification to an external party and then presenting the platform to them. The party would, in his own capability, try to navigate and find the functionality in the user interface and perform the corresponding action. When the action is completed and the desired outcome is presented according to specification and expectation is met, the requirement is verified.

### **3.6 Conclusion**

This chapter describes the design of the entire system. Functional and business requirements were set up to describe the goal of the software. The design procedure was thereafter described in further detail, and followed by the creation of both the front end and back end systems.

# CHAPTER 4

---

## Results

---

*'Persistence is very important. You should not give up unless you are forced to give up.'* -  
Elon Musk

## 4.1 Preamble

The previous chapter discussed the full design and implementation structure. After implementation, the system was deployed and ran live for 11 months and counting. A database dump was made to access the statistics for analysis further in this chapter.

This chapter will discuss the outcome of the tool after implementation and verify if the requirements were met. A survey and several case studies validate the tool. Lastly, metadata provides further insights from the task tool.

## 4.2 Software platform

A combination of the agile and iterative approach was followed, as discussed in Chapter 2, to create the back end as well as the front end of the design.

Figure 4.1 shows the entire software platform with the applicable tools and usages:

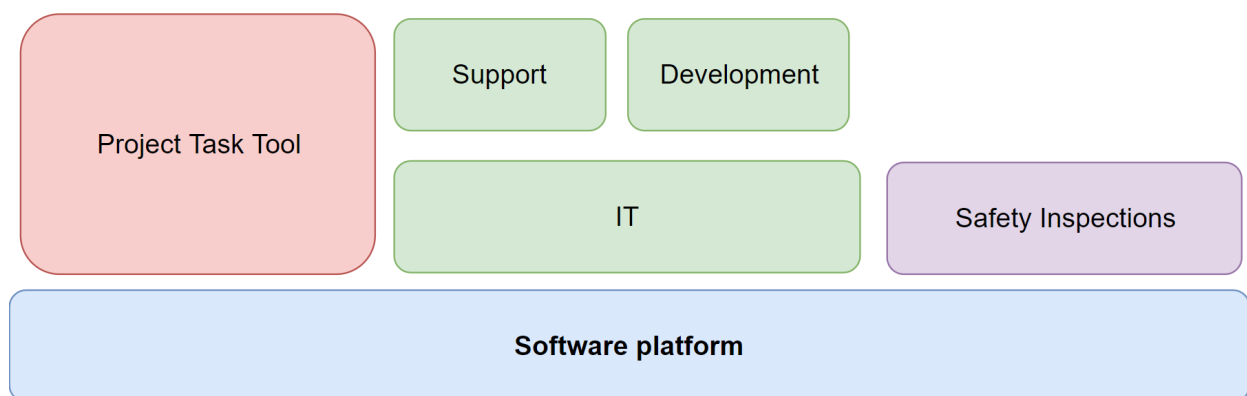


Figure 4.1: Software platform tools

The design of the software platform catered primarily for the project task tool. The tool is available to each user to use at own discretion. This enables users to collaborate, communicate and plan projects.

Another application that spawned soon after the release of the project task tool was the support tool. The support tool used the software platform to enable users in the company to log support tickets. The system notifies the IT and development departments.

The last application is safety inspections. A later discussion expands on a system to log inspections and potential safety problems.

## 4.3 Project task tool

Chapter 3 reflected on the design of the task tool and the implementation of each menu. The physical outcome is discussed in the forthcoming sections:

1. Task List
2. Create Task and Edit Task
3. Task Feed

### 4.3.1 Task list

Figure 4.2 shows the entry point of the user interface. A list of active tasks to which the user has either been linked to or created greets the user. The user can click on the task in the list and it will request the appropriate feed.

A short description of the critical points shown in Figure 4.2 follows:

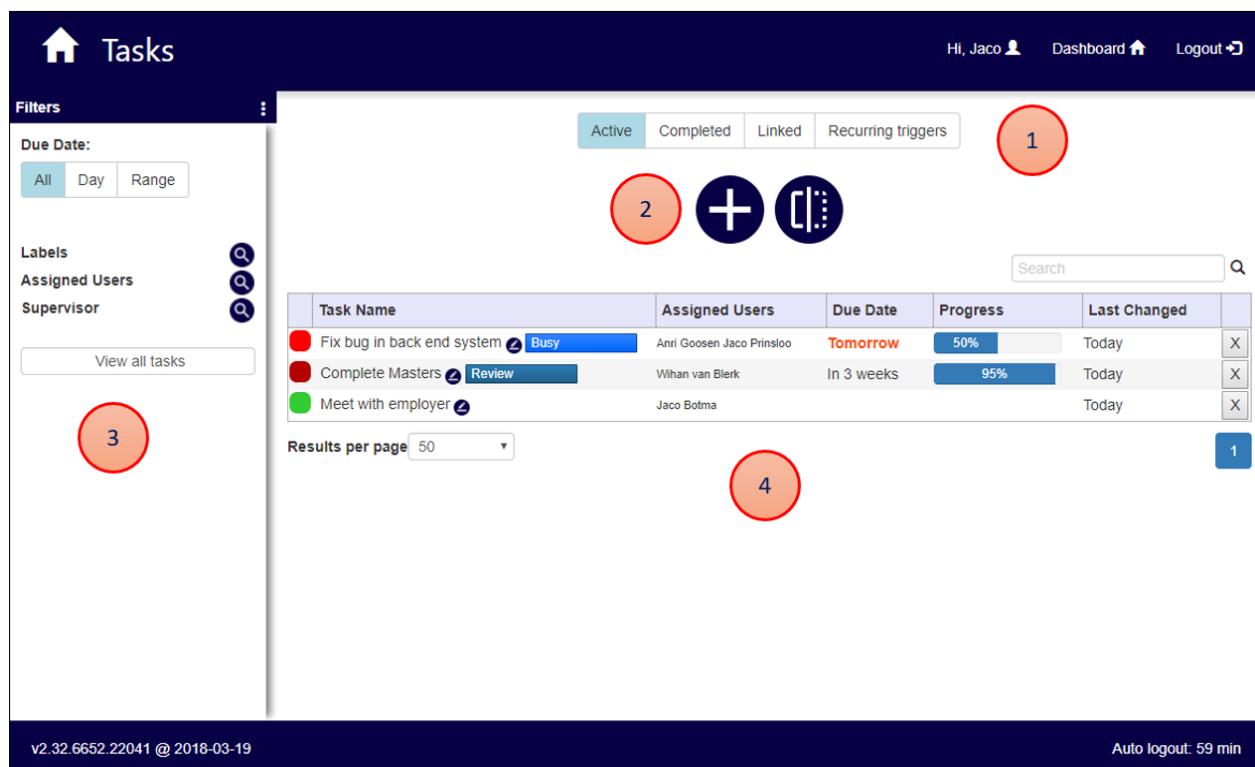


Figure 4.2: Task list

Each point correlates to a number in Figure 4.2:

1. Tabs that switch between the states of a task are allocated at the top of the page. 'Active' and 'Completed' states describe if the task is finished or still in progress. The linked tab indicates all the informed or assigned tasks for a user. 'Recurring triggers'

is a tab that groups tasks that have a recurring configuration together.

2. Centered action buttons draw attention to the goal of the page.

It allows the user to create a task and toggle additional information in the grid below the buttons. The additional information shows the date created, informed users, supervisor and who created the task.

3. The left side of the menu provides a filter. It allows filtering of tasks by the respective linked information or metadata.

If a user needs to quickly find a task according to high priority labels or by a particular user linked to a task quickly, the sidebar allows these filters to be applied.

4. The main task list is located in the center of the screen. The user can see at a glance the task name and some appropriate information linked to the task. This list links to the task feed for each task. Metadata such as the due date and progress on the task can be updated from the grid and allows ordering.

### 4.3.2 Create task

Figure 4.3 shows the menu that appears when a task has to be created. All configuration can initially be set up from here. Important points of the interface are described below:

The screenshot shows the 'Create Task' interface. At the top, there is a navigation bar with a back arrow, the title 'Create Task', and user information 'Hi, Jaco', 'Dashboard', and 'Logout'. Below the navigation bar, there are two circular icons: a save icon and a refresh icon. The main form contains the following fields and options:

- Task Name:** A text input field.
- Task Description (Optional):** A text area.
- Recurring Task Options (Optional):** A section with a minus sign icon, containing:
  - Time Frame:** A dropdown menu set to 'Weekly'.
  - Day of the week:** A dropdown menu set to 'Monday'.
  - Recur every:** A text input field set to '1' followed by 'week(s)'.
- Task creation time:** A section with a minus sign icon, containing:
  - HH mm:** A time picker set to '09 : 00'.
- Due Date (Optional):** A section with a minus sign icon, containing:
  - Time until due date:** A section with a minus sign icon, containing a 'DD' dropdown set to '2'.
  - List of next due dates:** A list of dates: 'Wed 2018-03-21 09:00', 'Wed 2018-03-28 09:00', 'Wed 2018-04-04 09:00', 'Wed 2018-04-11 09:00', and 'Wed 2018-04-18 09:00'.

Four red circles with numbers 1, 2, 3, and 4 highlight the save/refresh icons, the recurring task options section, the task creation time section, and the due date section respectively.

At the bottom of the interface, there is a footer with the version 'v2.32.6652.22041 @ 2018-03-19' and 'Auto logout: 58 min'.

Figure 4.3: Create and edit task interface

Each point correlates to a number in Figure 4.3:

1. Action buttons are again located at the top for consistency.

The user can save the task when it is fully configured. The action will route the user to the corresponding feed.

A reset action is also available. The task title and description are required to allow linked users to be knowledgeable on the subject.

2. Recurring options allow the user to create a template task that replicates itself with email notifications to remind the user of a point of interest. Figure 4.4 describes the configuration in more detail.

3. The 'task creation time' is required for all recurring tasks.

4. Due date calculations are provided to guide the user to execute the correct configuration.

Each time a specific configuration is selected and the due date is added, a list of examples is provided.

The algorithm used to calculate the due date in the front end is the same as the back end. It allows for consistency for testing purposes.

The recurring task allows the user to create a schedule of work. If a business has functions or essential dates that need particular attention, a task can be created preemptively to remind a person or a group of people.

The recurring task functions as a template for other tasks. It creates a duplicate of itself without the recurring configuration and comments. This type of task is used to remind the user of repeated work.

The figure displays four panels for configuring recurring tasks:

- Top Left (Daily):** Time Frame: Daily. Task creation time: 09:00. Day of the week: Monday, Tuesday, Wednesday, Thursday, Friday (checked); Saturday, Sunday (unchecked).
- Top Right (Weekly):** Time Frame: Weekly. Day of the week: Monday. Recur every 1 week(s). Task creation time: 09:00.
- Bottom Left (Monthly):** Time Frame: Monthly. Recur every 1 month(s). Task creation time: 09:00. A calendar for March 2018 is shown with the 21st selected.
- Bottom Right (Monthly):** Time Frame: Monthly. Task is created on e.g. 1st of each month. Recur every 1 month(s). Task creation time: 09:00.

Figure 4.4: Configuration of a recurring task

Figure 4.4 shows the possible time frames for a recurring task configuration. Several sections are created to cover a wide range of time interval categories.

- A daily configuration allows an option specific to days of the week.
- A monthly configuration allows the user to choose a specific day of the month.
- The user can choose a particular time for a specific day on a recurring task.
- Lastly the monthly configuration also allows complex combinations such as the first or last day of a month.

### 4.3.3 Edit and update task

The task feed interface provides a central hub for the aggregation of data around the task. The interface allows users to link each other to a task as Assigned, Informed or a Supervisor. It allows responsibility to be delegated to fit the current situation.

Below follows a short description of the possible actions on the task:

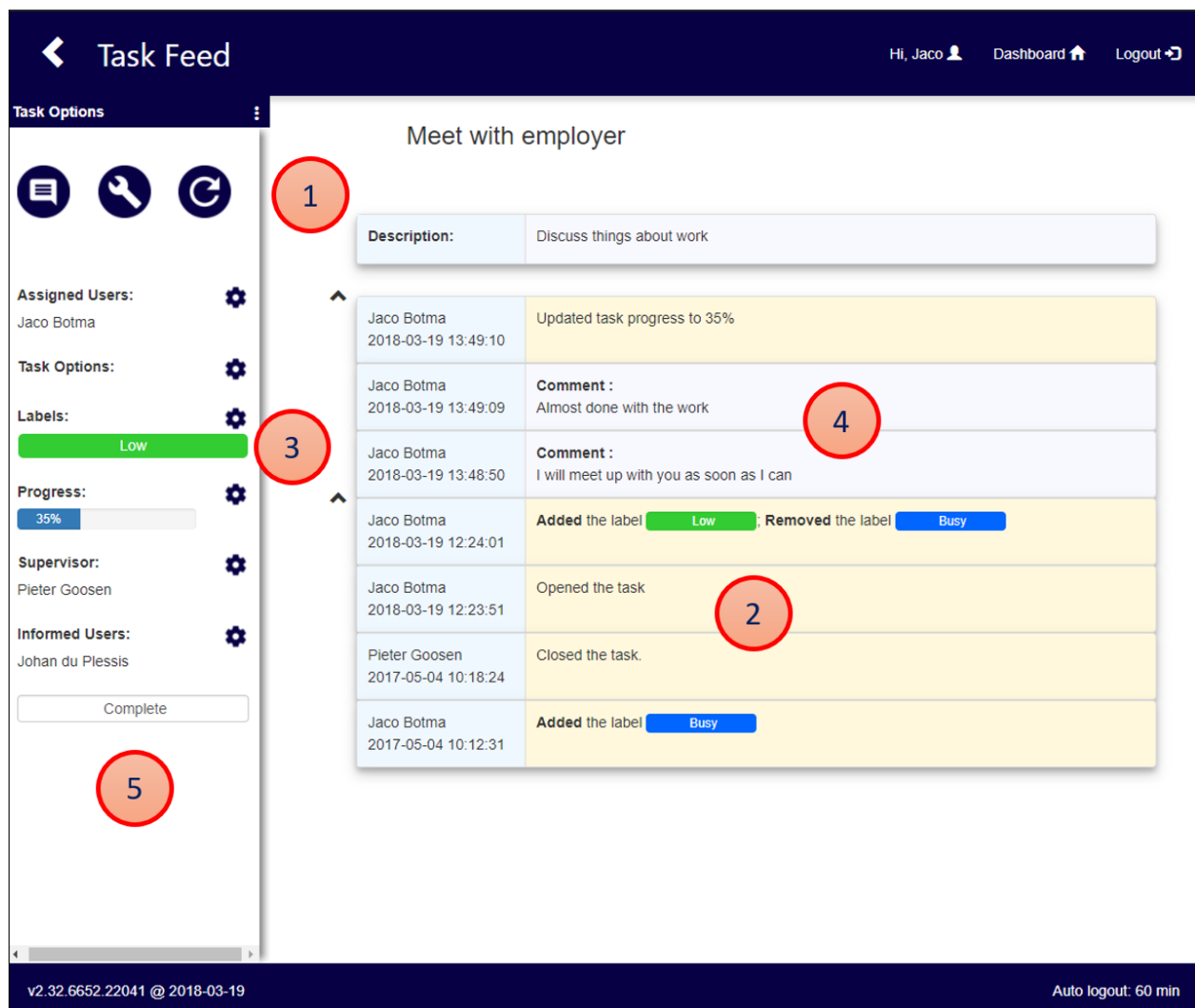


Figure 4.5: Task Feed Interface

1. An action button allows the user to comment, configure or refresh the task. Comments are a core part of teamwork and communication. Email functionality is added to notify users centrally.
2. A task change log is shown in the feed for record keeping and accountability. The system logs all changes to the task. The visibility of the information can be toggled to display only comments.
3. The interface allows linking of labels that show the status and priority. It allows categorisation of tasks and links importance or urgency value.
4. The feed list shows the task comments. The comments allow communication among users and function as a forum for an official record. The platform provides the ability to email other users linked to the task each time a user posts a new comment.
5. A single button provides open or close task functionality. The platform notifies all users via email once a user closes a task. The task is finally moved to the closed tab.

## 4.4 Verification

Verification ensures that the project meets all the requirements of Chapter 3. The business requirements are first discussed in Table 4.1 below. Most of the functionalities are verifiable through the user consensus of the survey discussed later. Tangible values are difficult to apply to the project efficiency due to the broad scope of projects and different people working on it. Another approach was thus taken to verify that the system works: asking the users.

Table 4.1: Verification of business requirements

<b>Business Requirements:</b>	<b>Way in which requirement is met through design and implementation</b>	
Generic engine. More than one business application.	The system is used as a ticketing system alongside the task system. This is evident from the case studies in a later section.	✓
Improve efficiency of internal processes.	From the survey, a consensus forms that the users feel that the efficiency is improved.	✓
Enforce planning through work delegation.	The tool provides a definite structure for planning with due dates, progress and labels. The users agreed that the tool delivers a planning platform by means of a survey.	✓

Table 4.2 describes the verification of the user requirements. Each menu in Chapter 3 had a specific design and had to accommodate these requirements through the design. The survey also validates some subjective requirements. The user requirements also link to the implemented functionality: if the functionality exists and physically works, the requirement is met.

Table 4.2: Verification of user requirements

<b>User Requirements</b>	<b>Way in which requirement is met through design and implementation</b>	
Search and filter task metadata.	Filters on the side bar, as well as search functionality is available from all menus where applicable.	✓

<b>User Requirements</b>	<b>Way in which requirement is met through design and implementation</b>	
Assign due dates for planning.	Due dates can be added and updated from both the task list and task edit menu.	✓
Categorise and prioritise the tasks.	Through the ability to label and measure progress of tasks a mechanism is provided to categorise and prioritise.	✓
Users only allowed certain actions per task.	Informed users and assigned users especially in IT support, are not able to unlink or label tasks. This is up to the IT manager.	✓

All of the functional requirements in Table 4.3 were met. Each requirement is in some form met through a menu or an action that can be accessed or modified.

The entire system with the implemented functionality was running in an active environment for at least 11 months after deployment.

Table 4.3: Verification of functional requirements

<b>Functional Requirements</b>	<b>Way in which requirement is met through design and implementation</b>	
Update title and description.	Available via task list and edit menu.	✓
Commenting must be allowed per task.	Task feed provides the ability to comment and entire history of comments for a specific task.	✓
Recurring options must be linked to the task.	A tab is available to configure recurring tasks in the task list. Recurring tasks were unit tested and are available from the create task menu.	✓
The task must have an open or closed status.	A tab is available in the task list for open and closed tasks. The task feed menu allows opening and closing of tasks.	✓
The platform must log each change to the task metadata with a date.	The database logs each change to the data automatically through procedures.	✓

Functional Requirements	Way in which requirement is met through design and implementation	
The task must have users linked to it with different roles.	Informed, Assigned and Supervisor roles are all able to be linked to users for tasks.	✓
The platform must log the creator of the task.	The task logs the created user automatically at creation and is visible through the task list.	✓
The system should provide a way to label tasks.	All changes to a task and its metadata are logged and shown in each feed.	✓

## 4.5 Validation: Case studies and survey

The system is developed to solve the multi-faceted problem explored in Chapter 1. Different case studies will be discussed to collect and evaluate the outcomes of the system. Each case study will be described and evaluated ending with a summary of the results found.

Key points of each case study will be evaluated:

1. A short description of the system. What the intentions of the system are, and if the system achieves the desired outcomes.
2. How the case study compares in a real-world application with the studies done in Chapter 2.
3. The requirement(s) that the case study satisfies, as specified in Chapter 3.

### 4.5.1 IT management

A requirement emerged from within the company that demanded a tighter grip on the IT department. Workflow can be logged and managed through the use of a ticketing system. The development team created a support tool soon after the initial release of the project task tool. The support tool allows users to log tickets online, after which the platform will notify the IT staff.

An additional branch stemmed from development support. This type of support is aimed at bug fixes and improvements for the integrated platform. It allows a transparent communication platform with a record of the work that was done. The employees create a prioritised to-do list available for the developers to see. Progress tracking is now possible where required meetings usually crunched time.

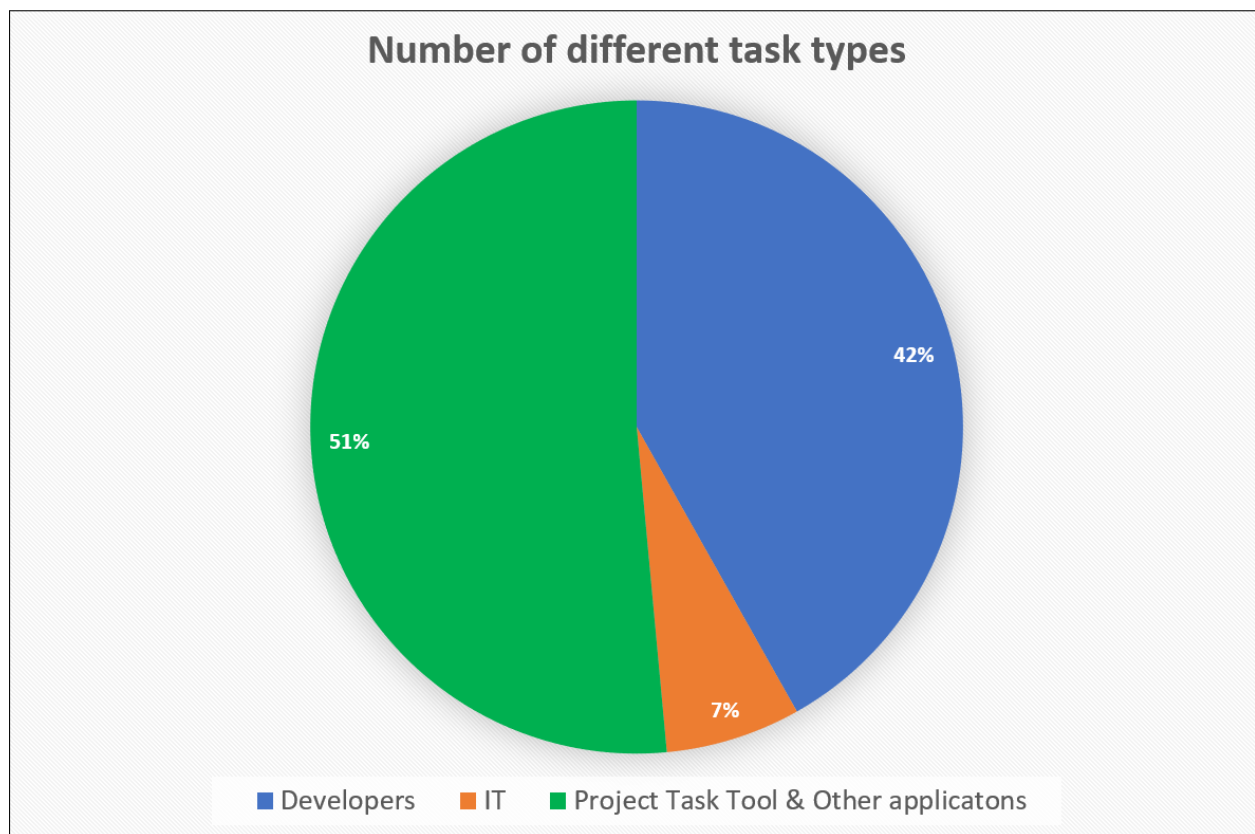


Figure 4.6: Percentage of each task type

Figure 4.6 was generated from a total of 2883 completed tasks. This high number proves that the company uses the system actively. The platform logged 193 IT support tickets, of which 181 were completed. From the development support side, 1013 tickets were logged with only 139 left open.

The ticketing software platform proves its business value and efficiency by actively quantifying the amount of work and logging all the tasks with its respective progress levels.

79.8% of all of the logged tasks, including support and project tasks, were completed at the time of writing this document <sup>1</sup>.

IT and development alone can boast an 87.4% completion rate (well above the total completion for the entire company). From the survey, available in the appendix, the IT managers would recommend the system, since it allows progress tracking and allows them to follow up on ongoing tasks.

<sup>1</sup> Queries and answers found in the appendix

The appendix also contains a list of example tickets logged by the users. This gives some insight into the type of support delivered to the company by the IT and development departments. It is found that the support tool targets specific areas mentioned in Chapter 2.

The first is human resource management where the tool assists with the automated management of the developments and IT departments. The second area in which the tool improves on is time efficiency. The tool helps effective management and presents a quantitative workflow to each developer or IT support staff member. The staff can now plan and target specific problems efficiently.

The tool lastly meets the business requirements in Chapter 3. This support tool is part of the generic software engine (requirement 1), and it enforces planning (requirement 3) through work delegation.

#### **4.5.2 Miscellaneous use cases**

Various other applications exist where the tasks system is used.

Applications for the tool prove the dexterous nature of the platform. The survey, discussed in a later section, as well as statistics drawn from task metadata, prove the value of the tool.

##### **Safety inspections**

An external company needed a ticketing system to log safety inspection data. The developed software platform was adapted for this specific situation. Additional metadata was linked to the task to fill in any specifications regarding the safety inspections.

Various categories contain hundreds of questions that the safety officers must answer during inspections. The company created 190 inspection tasks within a time span of a month. The platform generates reports from the task data. Each of these tasks has a set of questions linked to it.

Different types of inspections exist, with the answers linked to each question for the inspection. When an inspection identifies a dangerous work environment, a user is linked to the task to follow up on the inspection and sign off once the environment is safe again.

Through this methodology, the system keeps track of the person(s) responsible. In this

specific case study, the software platform is used to keep a record of all the entries and changes to a safety inspection.

The company uses this platform as a logging tool, as well as a management tool. This correlates with Sections 2.2.2 and 2.2.3. This application also satisfies business requirements 1 and 3 of Chapter 3. The system proves its value for yet another application by providing a versatile implementation interface.

### **Academic management**

The company used in this study, encourages further study at university. Lecturers are employed and supervise academic progress throughout the study of individual students.

At least 24% of the users of the platform agreed that they use tasks to manage and complete their work according to the survey discussed in a later section.

The lecturers use the task software platform as a way to manage the students' academic progress. Each student is encouraged to create a task. At each scheduled academic session, the students have to keep notes on the session on the task as proof of attendance.

### **Knowledge transfer: vacation work**

It is often found in practice that inexperienced workers only join the company over a vacation period due to their studies. Smaller, repetitive or less demanding tasks are assigned to these workers for completion.

This requires tight management to prevent workers wandering about the premises. 24% of the user group use the tool to assign work to their subordinates. A list of tasks is created and assigned to the vacation worker.

The platform creates a forum between the employee and employer. It allows the employee to ask questions and the employer to track progress.

### **Office admin**

The secretary also uses the project task tool. Any office admin is listed and completed. Many tasks cannot be completed initially due to external parties delaying processes. The system is used to keep track of ongoing events. Formal requests to employees are also available on a central communication platform. If feedback from a group of users attending an event is required for instance, a mechanism is available. 8% use the tool for admin, but almost 16%

use the tool for planning.

The office admin, vacation work and academic management prove that several real-world applications exist to satisfy the business requirements of Chapter 3. Close inspection of the literature in Sections 2.2.2 and 2.2.3 reveals that the case studies in Chapter 4 prove knowledge transfer and record keeping.

### 4.5.3 Survey

The system has to be measured and validated to prove that the software platform achieves the desired outcomes. The system only has value if it is used efficiently for its designed purpose. A survey was thus created to question the integrity and value of the system.

The survey targeted two focus groups on the employment ladder, namely the employees and the managers. The managers received different questions tailored to their positions.

The survey was designed with the intent of gathering a considerable amount of information from the users in the company without consuming time. From the start, an assumption could be made that not all users of the company will complete the survey. Data showed that approximately 89% of the company had used the tool in the past<sup>2</sup>.

The survey was designed in electronic format. It was also designed to assess the qualitative characteristics of the software. The tool provided immediate feedback on the results. These specifications cannot necessarily be measured in numeric values.

A set of close-ended questions were set up to be answered by the users of the system to evaluate whether the system serves its purpose qualitatively. The full survey, found in the appendix, lists the main information required:

1. Is the system integrated with the existing web platform?
2. Is the system easy to use and does it provide logical flow?
3. Does the feedback from the system allow you to prioritise and categorise your work?
4. Does the system provide a communication or feedback mechanism?
5. Does the system provide a record of the work done?
6. Can the system be used to provide knowledge transfer?

---

<sup>2</sup> Users with active accounts and access to the task tool

The survey was sent out to the specific users, and the results are discussed further on. The first topic the survey addresses is the diversity of projects for the employees. In Figure 4.7 below, the number of groups and the percentage of employees work in each group can be seen. This figure suggests that a comprehensive set of people working on several different areas use the tool within the company.

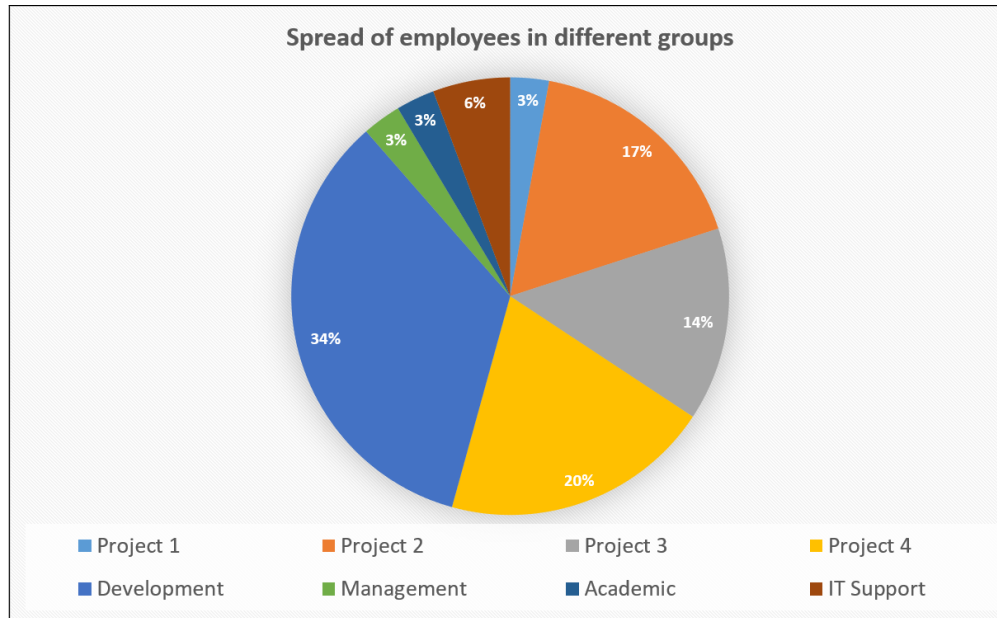


Figure 4.7: Spread of employees in different groups

The next question revealed what the primary use of the task tool is. Figure 4.8 shows that the primary use is development and support. The second use is as project task tool. This is due to the high demand for technical support within the company.

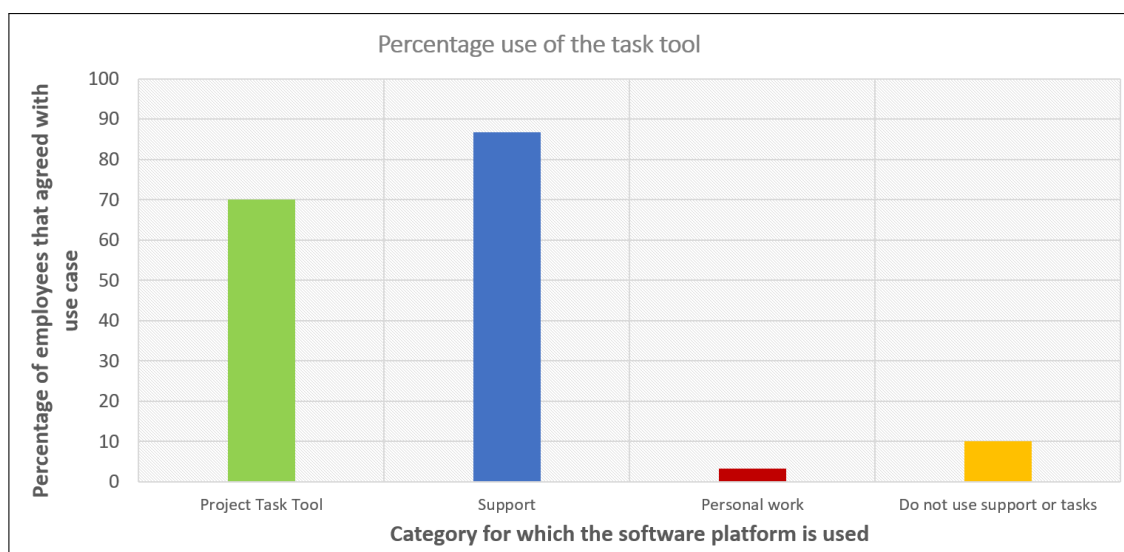


Figure 4.8: Percentage of people working in different groups using the task tool

Two separate questions were aimed at getting more insight on what Figure 4.8 represents. The following statistics suggest that the platform has multiple uses:

- 70% of the users who use the task tool, use it to log tickets.
- 41.4% receive work from their employers and 24.1% delegate work to their subordinates.
- 24.1% of the employees use the tool for personal use.

Verification and primarily the system validation came from the next set of questions. Each of the user requirements validated is found in Chapter 3.

Figure 4.9, on the next page, assigned a numeric value on how strongly the users agree with the design and functionality of the system. This figure illustrates that the active areas are work delegation and communication, but the knowledge transfer seems to meet less user satisfaction. Notice that the y-axis represents a user rating from 1 to 5, but was adjusted to start at 2.9 to give a better comparison perspective. The feedback provides a definite positive feel, with a lowest score of 3.9 and an average score of 4.14.

Further investigation revealed that the company uses the employees to retain knowledge instead of the tool. Section 2.2.3 describes the areas which lead to proper knowledge transfer. The company recognises the value of the knowledge due to the willingness to retain knowledge through the use of the tool. The company furthermore possesses the means for knowledge transfer.

The problem may lie with the willingness to learn new knowledge from the parties involved and the ability to learn. The use of the tool has not become part of the company culture and still needs work to prove useful for this purpose. On the other hand, the most favourable outlier is the general user consensus that the platform provides a central communication mechanism among users.

A critical area that improves efficient management is communication, as identified in Chapter 2. The users of the platform also agree that the system is integrated with existing platforms. This is a core part of the study. The design in Chapter 3 thus succeeded in its goal to integrate the platform.

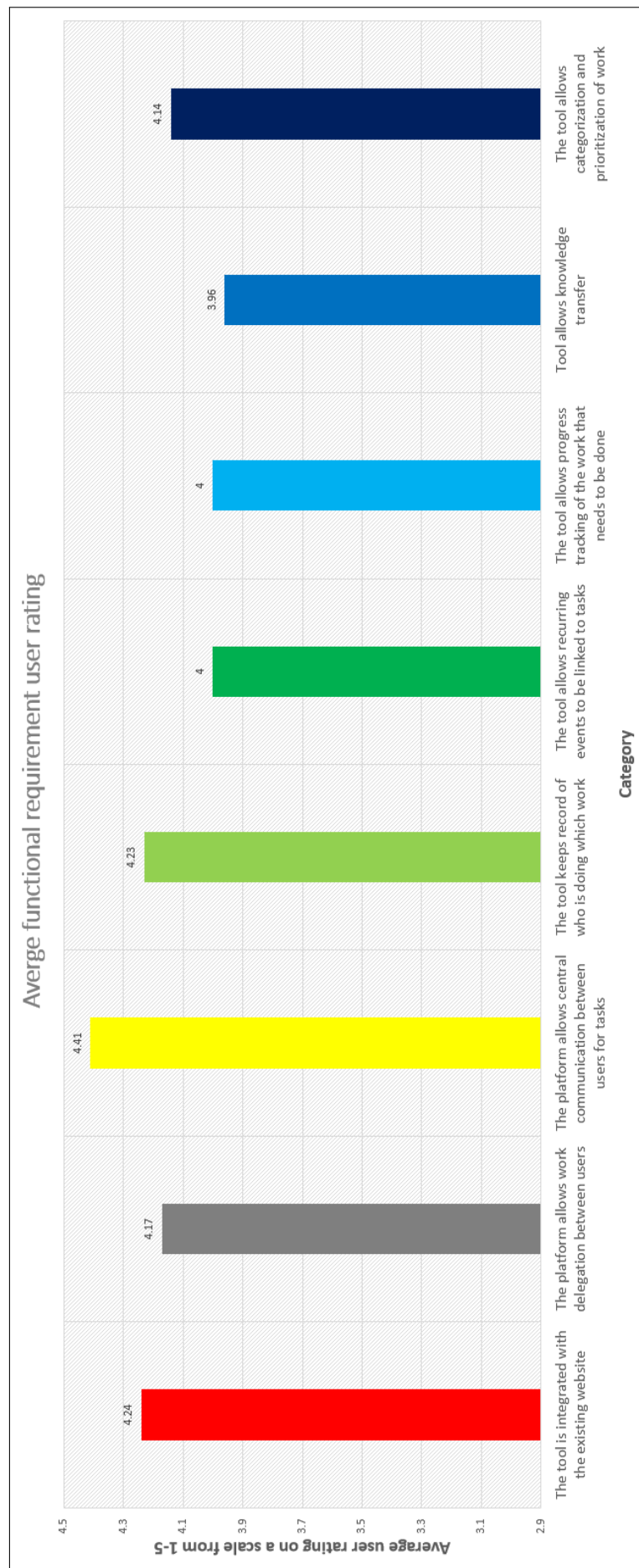


Figure 4.9: Average response of user agreement with functionality of the tool

## 4.6 Quantative results and general statistics

The software platform was developed with the goal of improving efficiency on new and existing projects. This is done by quantifying the amount of work done.

The company that implemented the software platform has several projects under management at any given time. Each project, with individual statistics, is discussed below:

1. The use of the system advances teamwork. The average number of linked users to a task.
2. If prioritisation takes place through the use of labels, how many labels are linked to tasks?
3. If the platform is used as a communication platform, how often comments are made?

To prove relevance in the conclusions drawn from the data set, an appropriate number of users must use the system. The initial rate of task generation is coupled with the rate at which the users adopted the system.

Figure 4.10 shows the rate of task generation by the company as a whole.

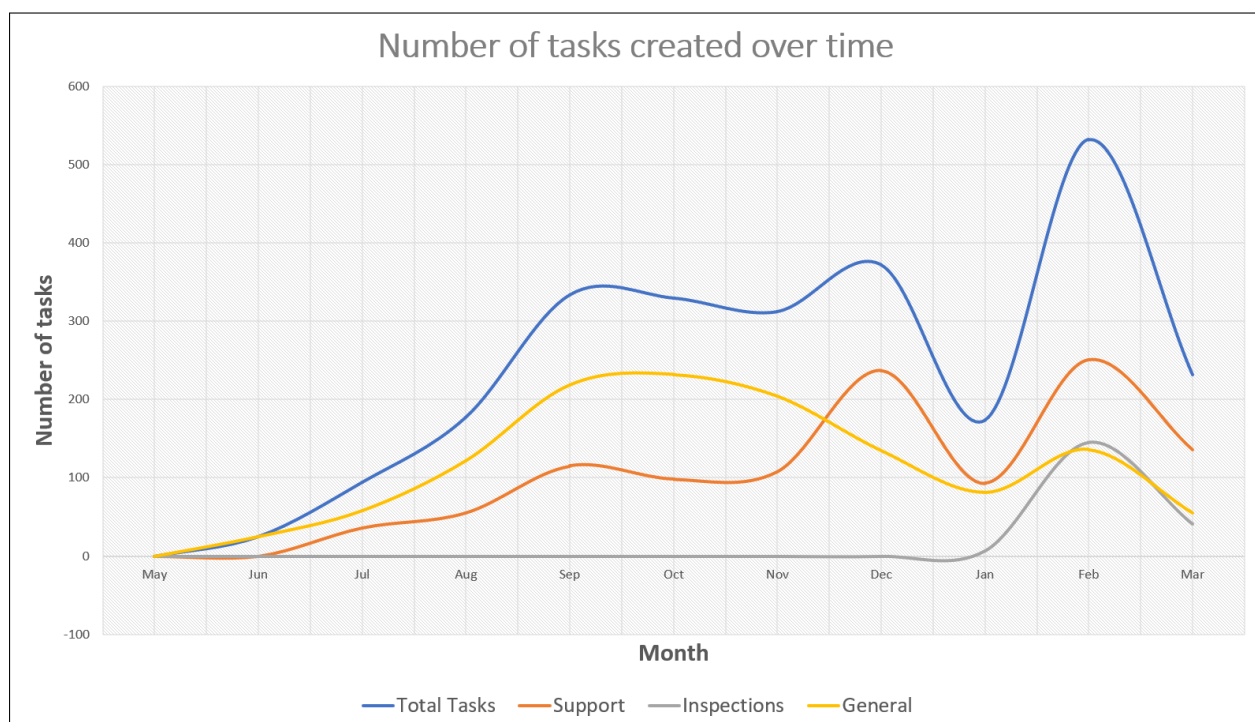


Figure 4.10: Creation of tasks over the months since platform deployment

A rate of almost three tasks per user is created per month. The system has thus been widely adopted and shows that the assumptions made from further statistics carry some

weight. The ‘Total Tasks’ represents a trend that indicates the cumulative of the other three trends (support, inspections and general). A decline in the use of the system is shown from December to January. This is due to the holiday season when workers usually take leave. The safety inspections tool was introduced from January to February. A sharp incline occurred in the cumulative number of tasks. Figure 4.10 shows a stable usage rate from August to November. The tool is thus actively used, and continuous development takes place.

#### 4.6.1 Linked users

An average of 2.5 users is linked to a task. This implies that the tool is used as a collaborative platform, since a considerable number more than one is using the tool.

On average 0.63 assigned users are linked and 1.73 informed. This implies that the task tool is used for informing a group of people on information more than work delegation.

Another statistic that needs to be noted is the average supervisors linked per task. 13 % of the tasks have a supervisor assigned. This gives an insight into the management structure of the company.

Most employees are assigned tasks either by personal or direct managed contact. It also implies that the employee takes own responsibility for work and that the superior may only monitor the progress of the task.

#### 4.6.2 Prioritisation and categorisation

The labels linked to the task indicate how the task is categorised. In Chapter 3 the purpose of the different types of labels was discussed in short. In Table 4.4 a summary of the different types of labels linked to the number of tasks can be found.

Table 4.4: Number of tasks linked to the different types of labels

Description text of the label	Type	Amount of Tasks
Low	Priority	109
Crisis	Priority	29
High	Priority	331
Medium	Priority	58
<b>Total</b>		<b>527</b>

Description text of the label	Type	Amount of Tasks
Busy	Status	282
Review	Status	45
Queued	Status	36
Testing	Status	55
Future	Status	79
Await Info	Status	70
Quick	Status	35
<b>Total</b>		<b>602</b>

Status labels are used more to tag the tasks with metadata. The user tag tasks with status labels and an almost even distribution lies within that category. The priority labels show that some are more used than others. A closer look gives a more in-depth insight into what the tasks are tagged with. A considerable number of tasks are tagged with ‘Busy’ and ‘Future’. This implies that the task tool is used for planning, as well as during the work process.

### 4.6.3 Communication

The communication of the task tool is done primarily through the commenting feature on the task list.

An average of 4.43 comments per task is created. This implies that besides a short description and title, further communication is necessary to enable an efficient workflow.

The average user comments 0.8 times on his task. This number reveals that the commenting functionality does not remain stagnant, but that there are more verbose users than others. The maximum number of comments on a task is up to 44 comments. Users engage in lengthy discussions, and the tool helps convey relevant information.

## 4.7 Summary of statistical results

Table 4.5 describes all the statistic values and their relevance to this chapter. The table indicates how the users use the tool.

Table 4.5: Descriptions and values of the different statistics in the chapter

Description	Result	Relevance
Project task tool and development and support percentage	51% task, 42% development and support	Indicates the spread of the project task tool usage.
Total number of tasks completed and open	2883 completed, 736 open	Total task completion rate of 79.8%
Number of support tickets and completed	193 created, 181 completed	The IT support has 93% completion rate.
Number of created and open development tickets with the completion rate	1013 created 139 open.	Development has a high rate of tickets and a completion rate of 86.2%
Number of inspection tickets created	190	The software platform rapidly provided a framework for a new application.
Percentage of users who use the tool to <b>complete their work</b>	24.1%	Supports the claim that the platform has multiple uses.
Percentage of users who use the tool to <b>supply subordinates with work</b>	24.1%	Supports the claim that the platform is used for work delegation.
Percentage of users who use the tool <b>for administration</b>	8%	Very few users use the framework for administration.
Percentage of users who use the tool <b>for planning</b>	16%	Supports the claim that the platform is used for planning.
Percentage of users who have <b>used the software platform</b>	89%	A wide range of users have found value in the software platform.

<b>Description</b>	<b>Result</b>	<b>Relevance</b>
Percentage of users who use the tool to <b>log tickets</b>	70%	A high number of users use the tool for support.
Percentage of users who <b>receive work from employer</b>	41.4%	A high number of users use the platform for work delegation.
Number of linked users on a task	2.5	Users use the platform for communication and work delegation.
Number of assigned and informed users on a task	0.63 assigned, 1.73 informed	The company uses the tool more for informing users than work delegation.
Percentage of tasks that have a supervisor	13%	The relatively low percentage suggests that employees take responsibility for their work.
The average number of comments per task.	4.43	A form of two-way communication takes place on average on every task.
The average number of comments of a user on his task.	0.8	Communication on the task takes place mostly on the side of external users linked to the task.
The maximum number of comments on a task	44	Lengthy discussions take place.

From table 4.5 a conclusion can be made that there are a variety of uses for the system. The tool is mainly used for task management and IT support. The task management allows the users to plan workloads and organise using labels. It also provides a communication platform that is actively used. The IT support workflow could now be measured. It indicates a high completion rate and efficiency in the company.

## 4.8 Conclusion

This chapter verifies and validates the software platform. The tool meets all the other requirements and reveals several small details about the operation of the company. The tool with its flexible back-end proves to be useful in various applications.

# CHAPTER 5

---

## Conclusion

---

*'Somewhere, something incredible is waiting to be known.'* - Carl Sagan

## 5.1 Discussion and conclusion

The efficient management of projects is a multi-faceted occurrence which has been investigated in many study areas. The industry created a way to manage projects and the human resources allocated to them through the use of task-based tools. These tools allowed planning, work delegation and communication to all the users of the system.

The literature describes many areas in which task-based tools give a business the upper hand in a competitive area. These benefits allow the company to plan and delegate work. Such a tool categorises the scope of a project and provides a record-keeping mechanism for knowledge transfer. The study suggested a broader scope of a task-based tool and developed a software platform for the tool as well.

The software platform would allow custom features and multiple tools to be based on the same code base. It would also put the company in the position to own the data generated by the tool. This study proposed business-, user- and functional requirements for a task tool and its software platform. The software platform would put a company in the position to reap all the aforementioned benefits.

A combination of the iterative and agile approaches was taken to build the software platform and the corresponding task tool on an ASP.NET web platform. An adapted MVC design pattern was used in an existing framework.

The study critically verified the functional requirements and validated the user and business requirements through the use of an electronic survey. The tool was, at the time of the study, actively used for multiple business applications. Statistically, the study showed that users use the tool for communication, work prioritisation and categorisation. Users agreed that the implemented task tool is integrated and addresses all the concerns from the study regarding the efficient management of projects. 70% use the platform to manage their projects and above 80% of the user base also use the software platform for IT and development support.

The software platform proved its business value through several case studies. The software platform was not only used for the project task tool but IT & development support as well as safety inspections. The users agreed that the software platform was integrated and the task tool served as a communication platform. The approval rating yielded above 4.14 out of 5 on average for all the project requirements. The users agreed that the tool helps them to

categorise work through the use of labels and to keep a record of employee responsibilities.

In conclusion, the software platform developed for this study proved that a task tool assists in the efficient management of projects.

## 5.2 Recommendations

From the survey additional responses were listed that provided valuable insight to the shortcomings of the tool:

1. Additional integration with an existing work log tool. This was suggested twice by different users. This is possible due to the work log and task tool existing in the same platform. The integration would be able to link the amount of hours spent on a task to an appropriate work log. The actual cumulative amount of hours per task could thus be calculated. This would enable a more analytical insight to how time is spent in the company.
2. The tool sends out emails as notification method. These mails are ignored in some cases by the user, since these are automatically generated. The manager thus has to manually remind the employee to complete the work.

Adoption of the tool in the company is part of the education process. The use of the tool should be part of the internal work processes a company to reap maximum benefit. Users should be motivated and informed on the different functionalities and benefits of using the tool.

3. Personal reporting should be available on the tasks per user. It would allow an analytical view to enable the user to make informative decisions on how he spends his time.

The recommendation is in the same line as linking with the work log tool. A definite need exists to quantify the amount of time spent on the work.

4. A project board linking certain tasks to certain projects will add a project-wide view of occurrences in the company. It will further allow tasks to be classified and give users a broad overview of the progress of a project.

Recommendations for the literature study would include:

1. Research on the effect the long-term use of such a system has on the employees and procedures in an organisation. Chapter 2 addressed several articles that investigate how to remediate inefficiencies in a company [5, 60]. Several remediations are addressed by the software platform. There is however, no ‘before’ and ‘after’ comparison available in

a specific case study over a long-term period. Additional research could provide insight on if such a solution would be beneficial or hinder the effectiveness of a company.

2. The financial impact has of such a tool due to the use of the software platform needs further investigation [10]. Similar systems have been created that manage the efficiency in projects [8, 9]. Further investigation of the financial return on investment would deliver meaningful insight on the actual value of such a system. This would require the software platform to capture hourly data and the study to have some form of access to the finances of a company.

---

## Bibliography

- [1] N. Carbonara and R. Pellegrino, “Public-private partnerships for energy efficiency projects: A win-win model to choose the energy performance contracting structure,” *Journal of Cleaner Production*, vol. 170, pp. 1064–1075, 2018.
- [2] L. A. Ika, “Project success as a topic in project management journals.” *Project Management Journal*, vol. 40, no. 4, pp. 6 – 19, 2009. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=45361822&site=eds-live>
- [3] F. T. Berssaneti and M. M. Carvalho, “Identification of variables that impact project success in Brazilian companies,” *International Journal of Project Management*, vol. 33, no. 3, pp. 638–649, 2015.
- [4] A. J. Shenhar and O. Levy, “Mapping the dimensions of project success.” *Project Management Journal*, vol. 28, no. 2, p. 5, 1997. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=94345&site=eds-live>
- [5] A. Pak, L. L. Carden, and J. V. Kovach, “Integration of project management, human resource development, and business teams: A partnership, planning model for organizational training and development initiatives.” *Human Resource Development International*, vol. 19, no. 3, pp. 245 – 260, 2016. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=psych&AN=2016-27071-005&site=eds-live>
- [6] W. Mainga, “Examining project learning, project management competencies, and project efficiency in project-based firms (pbfs),” *International Journal of Managing Projects in Business*, vol. 10, no. 3, pp. 454–504, 2017.
- [7] J. V. Chen, I.-H. Lu, D. C. Yen, and A. E. Widjaja, “Factors affecting the performance of internal control task team in high-tech firms,” *Information Systems Frontiers*, vol. 19, no. 4, pp. 787–802, 2017.
- [8] H. A. Johnson, “Trello.” *Journal of the Medical Library Association*, vol. 105, no. 2, pp. 209 – 211, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=c8h&AN=122316226&site=eds-live>

- 
- [9] R. Featherstone, “Electronic resources reviews. basecamp.” *Journal of the Medical Library Association*, vol. 97, no. 1, p. 67, 2009. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=c8h&AN=105357939&site=eds-live>
- [10] K. Artto, A. Valtakoski, and H. Kärki, “Organizing for solutions: How project-based firms integrate project and service businesses,” *Industrial Marketing Management*, vol. 45, pp. 70–83, 2015.
- [11] H. Yang, “Impact of time management for it services management,” *Procedia Computer Science*, vol. 91, pp. 700–706, 2016.
- [12] E. J. Jeong, J. H. Bae, and S. R. Jeong, “Guidelines aimed at reducing the risks of user acceptance delay in the context of an it service project management plan,” *International Journal of Electrical and Computer Engineering*, vol. 5, no. 4, p. 832, 2015.
- [13] L. Argote and P. Ingram, “Knowledge transfer : A basis for competitive advantage in firms,” *Organizational behavior and human decision processes*, vol. 82, no. 1, pp. 150–169, 2000.
- [14] J. Emblemståg, “Handling risk uncertainty in project planning.” *Journal of Modern Project Management*, pp. 20 – 31, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=120752319&site=eds-live>
- [15] B. MUSKAT and M. DEERY, “Knowledge transfer and organizational memory: An events perspective.” *Event Management*, vol. 21, no. 4, pp. 431 – 447, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=hjh&AN=125478663&site=eds-live>
- [16] E. K. Melton and K. J. Meier, “For the want of a nail: The interaction of managerial capacity and human resource management on organizational performance.” *Public Administration Review*, vol. 77, no. 1, pp. 118 – 130, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=120437187&site=eds-live>
- [17] T. Reunanen, “Human factor in time management,” *Procedia Manufacturing*, vol. 3, pp. 709–716, 2015.

- 
- [18] C. A. Gorman, J. P. Meriac, S. G. Roch, J. L. Ray, and J. S. Gamble, “An exploratory study of current performance management practices: Human resource executives’ perspectives,” *International Journal of Selection and Assessment*, vol. 25, no. 2, pp. 193–202, 2017.
- [19] J. Korff, T. Biemann, and S. C. Voelpel, “Human resource management systems and work attitudes: The mediating role of future time perspective,” *Journal of Organizational Behavior*, vol. 38, no. 1, pp. 45–67, 2017.
- [20] R. Flores-Fillol, S. Iranzo, and F. Mane, “Teamwork and delegation of decisions within the firm,” *International Journal of Industrial Organization*, vol. 52, pp. 1–29, 2017.
- [21] D. Walker and B. Lloyd-Walker, “Understanding collaboration in integrated forms of project delivery by taking a risk-uncertainty based perspective.” *Administrative Sciences (2076-3387)*, vol. 6, no. 3, pp. 1 – 17, 2016. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=118306843&site=eds-live>
- [22] A. M. Kalimullin, V. T. Yungblud, and E. A. Khodyreva, “Characteristic features of innovation project management aimed at university human resource development.” *International Journal of Environmental and Science Education*, vol. 11, no. 9, pp. 2237–2253, 2016.
- [23] P. Agarwal and S. Sajid, “A study of job satisfaction, organizational commitment and turnover intention among public and private sector employees.” *Journal of Management Research (09725814)*, vol. 17, no. 3, 2017.
- [24] J. Bérubé and J.-B. Gauthier, “Compromise between creative activities and project management activities: a contingency factor,” *The Journal of Modern Project Management*, vol. 5, no. 2, 2017.
- [25] J. K. Pinto and D. P. Slevin, “Project success: definitions and measurement techniques.” Project Management Institute, 1988.
- [26] E. Barden, “Don’t go chasing waterfalls.” *NZ Business + Management*, p. 23, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=f5h&AN=124766321&site=eds-live>
- [27] B. B. Shao, P.-Y. Yin, and A. N. Chen, “Organizing knowledge workforce for specified iterative software development tasks,” *Decision Support Systems*, vol. 59, pp. 15–27, 2014.

- 
- [28] M. Adam Van Albrecht and R. Nemani, “Documenting agile project knowledge: A review of knowledge capture for agile practices,” *International Journal of Science, Engineering and Computer Technology*, vol. 4, no. 6, p. 194, 2014.
- [29] A. Izmailov, D. Korneva, and A. Kozhemiakin, “Effective project management with theory of constraints,” *Procedia-Social and Behavioral Sciences*, vol. 229, pp. 96–103, 2016.
- [30] F. Niederman, B. Müller, and S. T. March, “Using process theory for accumulating project management knowledge: A seven-category model.” *Project Management Journal*, vol. 49, no. 1, pp. 6 – 24, 2018. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=127543131&site=eds-live>
- [31] J. Beach and J. Oates, “Maintaining best practice in record-keeping and documentation,” *Nursing Standard*, vol. 28, no. 36, pp. 45–50, 2014.
- [32] J. Söderlund, A. L. Vaagaasar, and E. S. Andersen, “Relating, reflecting and routinizing: Developing project competence in cooperation with others,” *International Journal of Project Management*, vol. 26, no. 5, pp. 517–526, 2008.
- [33] J. L. Cummings and B.-S. Teng, “Transferring R&D knowledge: the key factors affecting knowledge transfer success,” *Journal of Engineering and Technology Management*, vol. 20, no. 1-2, pp. 39–68, 2003.
- [34] K. Ayas, “Integrating corporate learning with project management.” *International Journal of Production Economics*, vol. 51, no. 1/2, pp. 59 – 67, 1997. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=11945242&site=eds-live>
- [35] D. Zhao, M. Zuo, and X. N. Deng, “Examining the factors influencing cross-project knowledge transfer: An empirical study of it services firms in China,” *International Journal of Project Management*, vol. 33, no. 2, pp. 325–340, 2015.
- [36] D. J. Snowden and M. E. Boone, “A leader’s framework for decision making. (cover story).” *Harvard Business Review*, vol. 85, no. 11, pp. 68 – 76, 2007. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=27036324&site=eds-live>

- 
- [37] a. George P. Huber, "Organizational learning: The contributing processes and the literatures." *Organization Science*, no. 1, p. 88, 1991. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsjsr&AN=edsjsr.2634941&site=eds-live>
- [38] L. S. Lai, "A synergistic approach to project management in information systems development," *International Journal of Project Management*, vol. 15, no. 3, pp. 173–179, 1997.
- [39] N. Jyothi and A. Parkavi, "A study on task management system," in *Research Advances in Integrated Navigation Systems (RAINS)*, *International Conference on*. IEEE, 2016, pp. 1–6.
- [40] J. Gwizdka and M. Chignell, "Individual differences and task-based user interface evaluation: a case study of pending tasks in email," *Interacting with computers*, vol. 16, no. 4, pp. 769–797, 2004.
- [41] E. N. Márquez, "Excel-based workbook for quality management (ewqm)," *Procedia-Social and Behavioral Sciences*, vol. 174, pp. 298–304, 2015.
- [42] K. Smith, "Managing electronic resource workflows using ticketing system software," *Serials Review*, vol. 42, no. 1, pp. 59–64, 2016.
- [43] A. G. Silvius and C. M. Silvius, "Exploring functionality of mobile applications for project management," *Procedia Computer Science*, vol. 64, pp. 343–351, 2015.
- [44] M. van der Bank, "A modular framework for the effective development of web-based applications," Master's thesis, North-West University, South-Africa, 2017.
- [45] D. J. Solove and D. K. Citron, "Risk and anxiety: A theory of data-breach harms." *Texas Law Review*, vol. 96, no. 4, pp. 737 – 786, 2018. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=128451221&site=eds-live>
- [46] Y. QING, C. RONG, and X. XIAOBING, "Consistency between consumer personality and brand personality influences brand attachment." *Social Behavior Personality: an international journal*, vol. 43, no. 9, pp. 1419 – 1427, 2015. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=s3h&AN=110384240&site=eds-live>

- 
- [47] J. Heagney, *Fundamentals of project management*. AMACOM Div American Mgmt Assn, 2016.
- [48] M. Ringis and S. Bērziša, “Efficiency measurement of project management software usage at state social insurance agency.” *Information Technology Management Science*, vol. 19, no. 1, pp. 65 – 70, 2016. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=120842144&site=eds-live>
- [49] Y. K. Z. Teslia, Yehorchenkova and Khlevna, “Development of principles and method of electronic project management.” *Eastern-European Journal of Enterprise Technologies*, vol. 89, no. 3, pp. 23 – 29, 2017. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=126046603&site=eds-live>
- [50] S. Srivannaboon and D. Z. Milosevic, “A two-way influence between business strategy and project management,” *International Journal of Project Management*, vol. 24, no. 6, pp. 493–505, 2006.
- [51] R. Atkinson, “Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria,” *International journal of project management*, vol. 17, no. 6, pp. 337–342, 1999.
- [52] A. J. Shenhar and D. Dvir, *Reinventing project management: the diamond approach to successful growth and innovation*. Harvard Business Review Press, 2007.
- [53] G. Purna Sudhakar, “A model of critical success factors for software projects,” *Journal of Enterprise Information Management*, vol. 25, no. 6, pp. 537–558, 2012.
- [54] O. Motaleb and M. Kishk, “An investigation into the risk of construction project delays in the uae.” *International Journal of Information Technology Project Management, Volume 4 Number 3*, 2013. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsoai&AN=edsoai.891551596&site=eds-live>
- [55] A. El Yamami, S. Ahriz, K. Mansouri, M. Qbadou, and E. H. Illousamen, “Representing it projects risk management best practices as a metamodel,” *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2062–2067, 2017.
- [56] C. Desmond, “Project management tools beyond the basics,” *IEEE Engineering Management Review*, vol. 45, no. 3, pp. 25–26, 2017.

- 
- [57] —, “Project management tools software tools,” *IEEE Engineering Management Review*, vol. 45, no. 4, pp. 25–25, 2017.
- [58] S. Karkouliau, G. Assaker, and R. Hallak, “An empirical study of 360-degree feedback, organizational justice, and firm sustainability,” *Journal of Business Research*, vol. 69, no. 5, pp. 1862–1867, 2016.
- [59] T. R. Browning, “Managing complex project process models with a process architecture framework,” *International Journal of Project Management*, vol. 32, no. 2, pp. 229–241, 2014.
- [60] A. Keegan, C. Ringhofer, and M. Huemann, “Human resource management and project based organizing: Fertile ground, missed opportunities and prospects for closer connections,” *International Journal of Project Management*, vol. 36, no. 1, pp. 121–133, 2018.
- [61] M. Foucault, C. Teyton, D. Lo, X. Blanc, and J.-R. Falleri, “On the usefulness of ownership metrics in open-source software projects,” *Information and Software Technology*, vol. 64, pp. 102–112, 2015.
- [62] L. Carden and T. Egan, “Does our literature support sectors newer to project management? The search for quality publications relevant to non-traditional industries,” *Project Management Journal*, vol. 39, no. 3, pp. 6 – 27, 2008.
- [63] W. R. Duncan, “A guide to the project management body of knowledge,” 1996.
- [64] A. Singh and P. J. Kaur, “A simulation model for incremental software development life cycle model,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 7, 2017.
- [65] A. S. Campanelli, R. D. Camilo, and F. S. Parreiras, “The impact of tailoring criteria on agile practices adoption: A survey with novice agile practitioners in Brazil,” *Journal of Systems and Software*, vol. 137, pp. 366–379, 2018.
- [66] M. del Pilar Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. L. Cuadrado, “Analyzing best practices on web development frameworks: The lift approach,” *Science of Computer Programming*, vol. 102, pp. 1–19, 2015.
- [67] S. Tilkov and S. Vinoski, “Node. js: Using javascript to build high-performance network programs,” *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.

- [68] A. Sarkar and N. C. Debnath, “Business-object oriented requirements engineering framework.” *Journal of Computational Methods in Sciences & Engineering*, vol. 12, pp. 39 – 51, 2012. [Online]. Available: <http://nwulib.nwu.ac.za/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=83380795&site=eds-live>
- [69] J. Herman, “A centralised data management system for the South African mining and industrial sectors,” Master’s thesis, North-West University, South-Africa, 2017.

# APPENDIX I

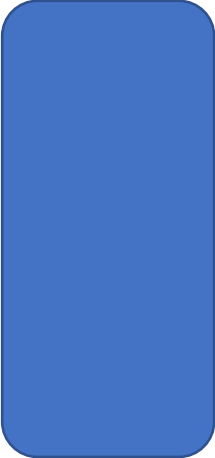
---

## Project task tool survey

## Tasks & Support Tool Survey

To generate some Qualitative results for my master's, would you please be so kind to fill in the survey.

Under which project or group do you currently operate? \*



Other...

For what purpose do you use the task/support tool? \*

- Tasks for work
- Log tickets for IT and development support
- I do not use tasks or support
- I use tasks for personal work

If you use tasks/support for work, how do you use the platform?

- My manager gives me tasks to complete
- I log IT or support tickets
- I use tasks to manage and complete my own work
- I use it to ensure subordinates receives work requirements/tasks

## If you use tasks for personal use, how do you use the platform?

- I use it for my masters or for academic purposes
- I use it to maintain my office or personal admin
- I use the tool for personal projects
- I use the tool to inform other people about events, social, work or meetups etc.
- I use it to plan my work or plan events
- No personal use

## Task system features

Description (optional)

The tool is integrated with the existing website



	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The platform allows work delegation between users.

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The platform allows central communication between users for tasks.

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The tool keeps record of who is doing which work. \*

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The tool allows recurring events to be linked to tasks.

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The tool allows progress tracking of the work that needs to be done.

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

Use tasks to recall previous information on work done. The list of tasks per user is used as a reference for other that will be doing the same work. (Knowledge transfer)

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

The tool allows categorization and prioritization of work

	1	2	3	4	5	
Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Agree

Would you say you reap benefit from using this tool?

- Yes
- No

## Managers Only\*

You may skip this question if you do not use the tool to manage your employees

How does the task tool influence your management?

- The task tool allows me to plan and delegate time to certain tasks.
- I have more control over the work delegation for the employees.
- I have more information on the progress of my employees.
- The tool allows me to prioritize important parts of the project.
- The tool keeps record and gives responsibility to the employees.
- The tool is used for knowledge transfer. I.E. tasks from a previous user is speculated or assigned to a new user.

Are there any suggestions on how to improve tasks or support tool? Any new features or requirements would be appreciated?

Long answer text

---

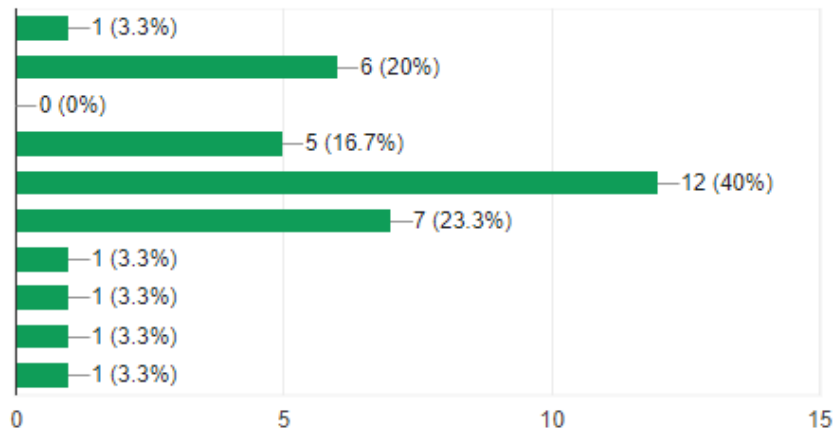
# Tasks & Support Tool Survey

30 responses

[Publish analytics](#)

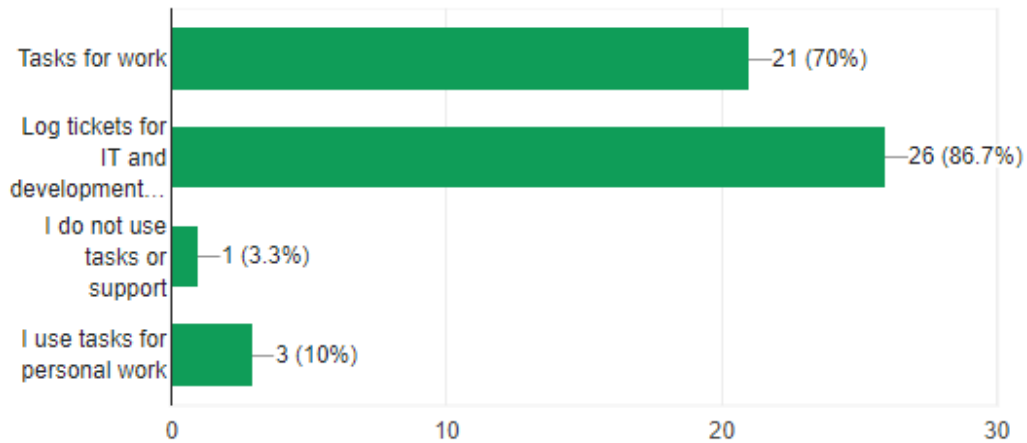
## Under which project or group do you currently operate?

30 responses



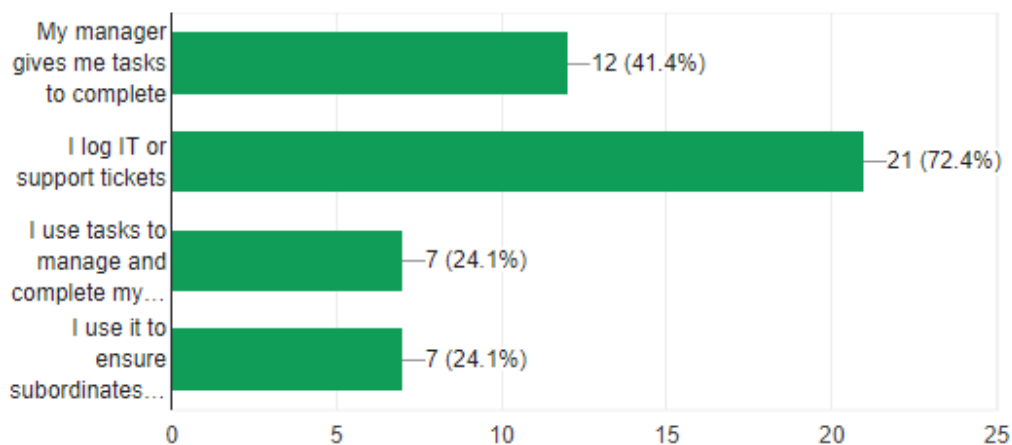
## For what purpose do you use the task/support tool?

30 responses



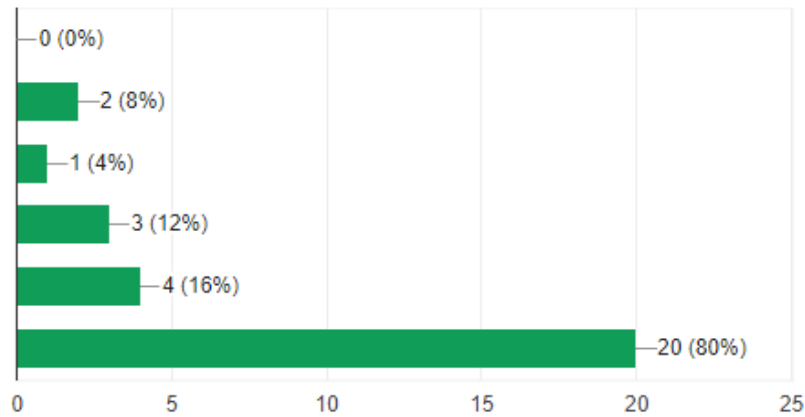
## If you use tasks/support for work, how do you use the platform?

29 responses



## If you use tasks for personal use, how do you use the platform?

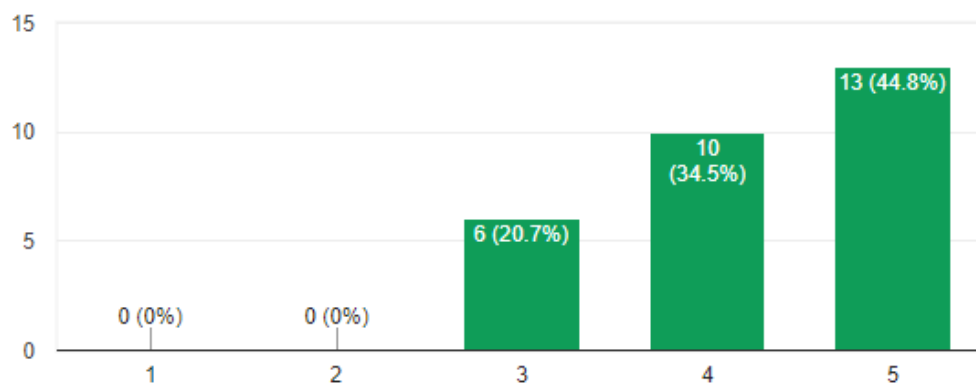
25 responses



## Task system features

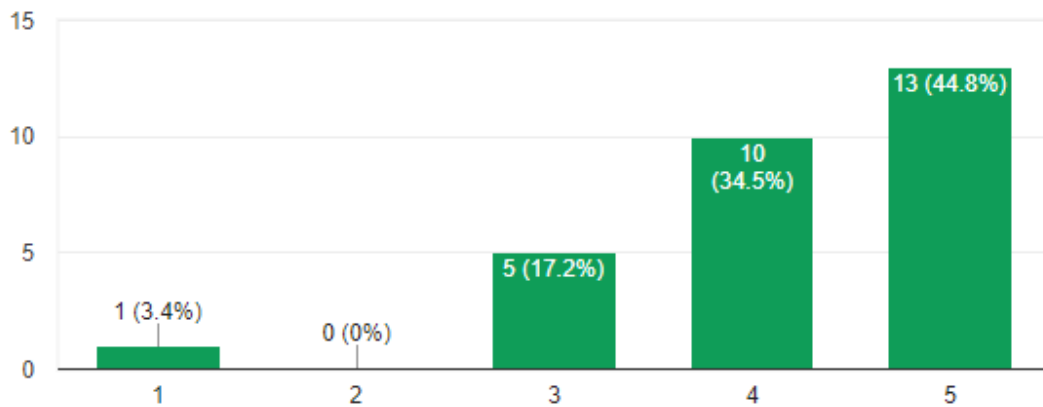
### The tool is integrated with the existing website

29 responses



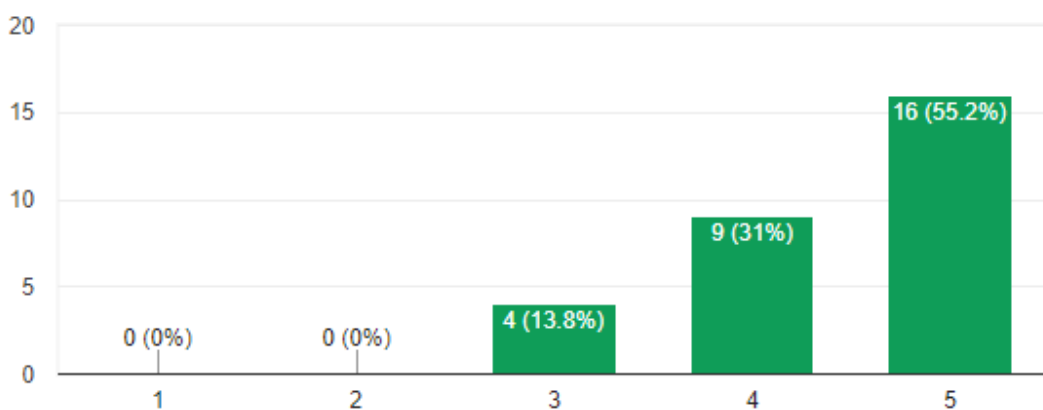
### The platform allows work delegation between users.

29 responses



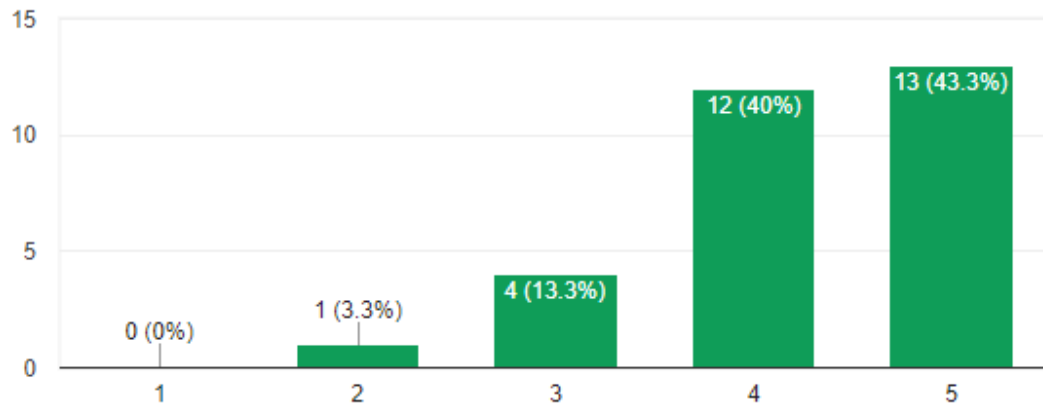
### The platform allows central communication between users for tasks.

29 responses



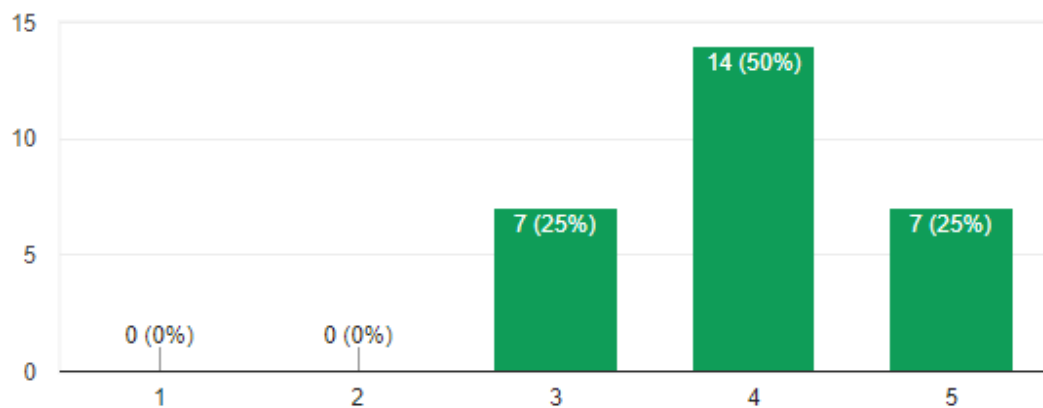
### The tool keeps record of who is doing which work.

30 responses



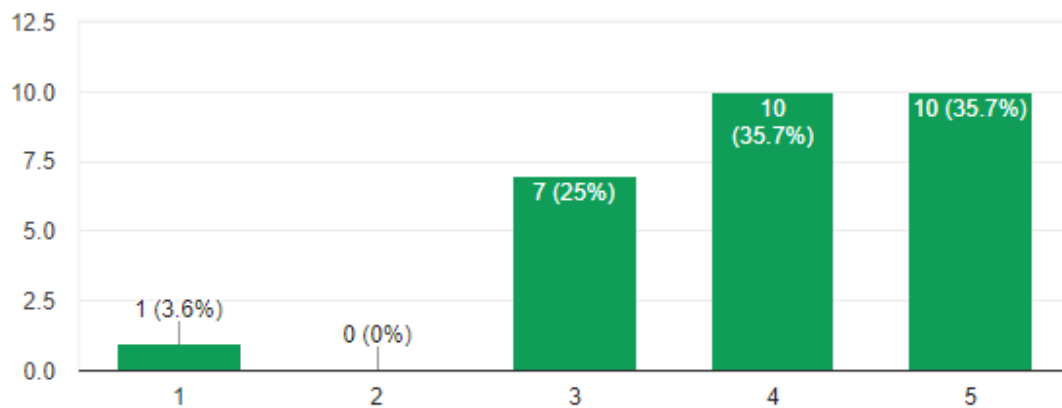
### The tool allows recurring events to be linked to tasks.

28 responses



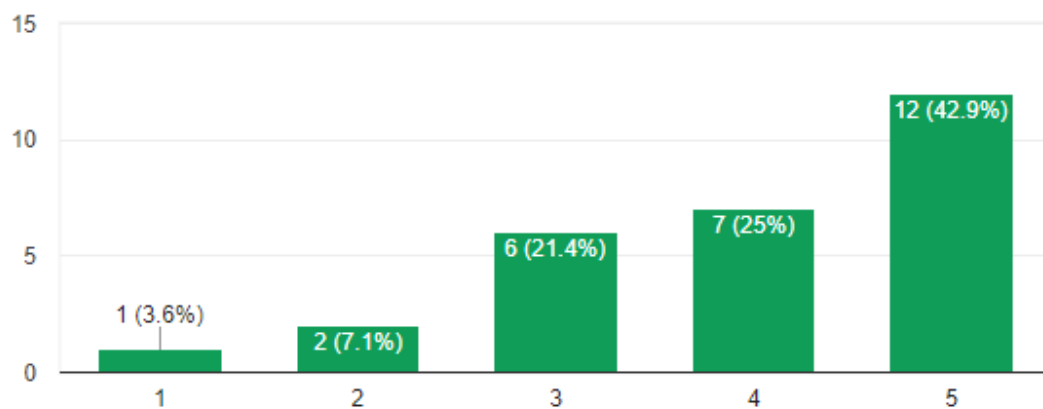
The tool allows progress tracking of the work that needs to be done.

28 responses



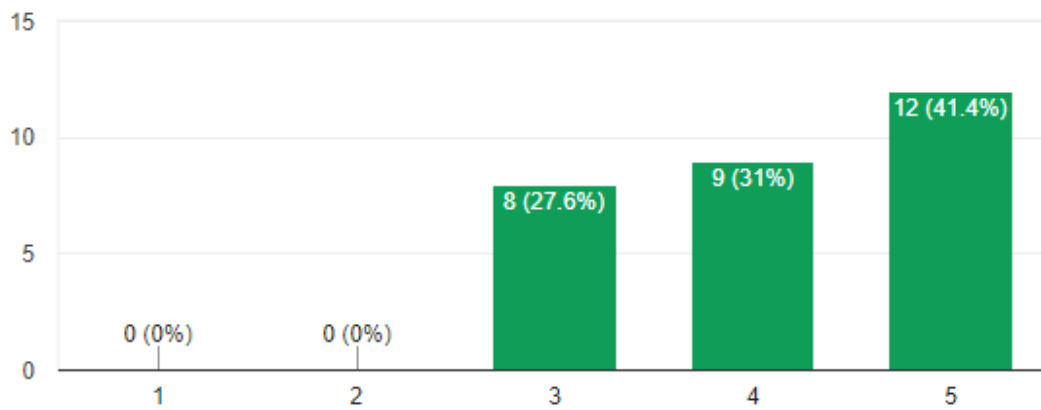
Use tasks to recall previous information on work done. The list of tasks per user is used as a reference for other that will be doing the same work. (Knowledge transfer)

28 responses



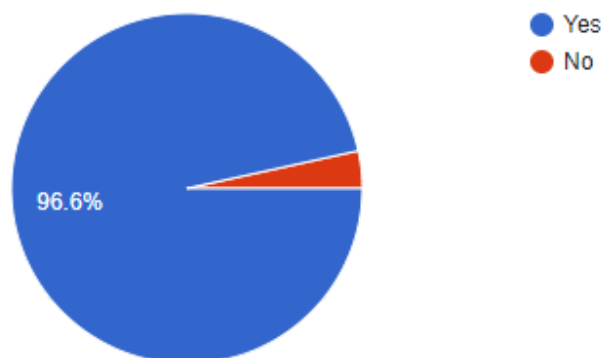
## The tool allows categorization and prioritization of work

29 responses



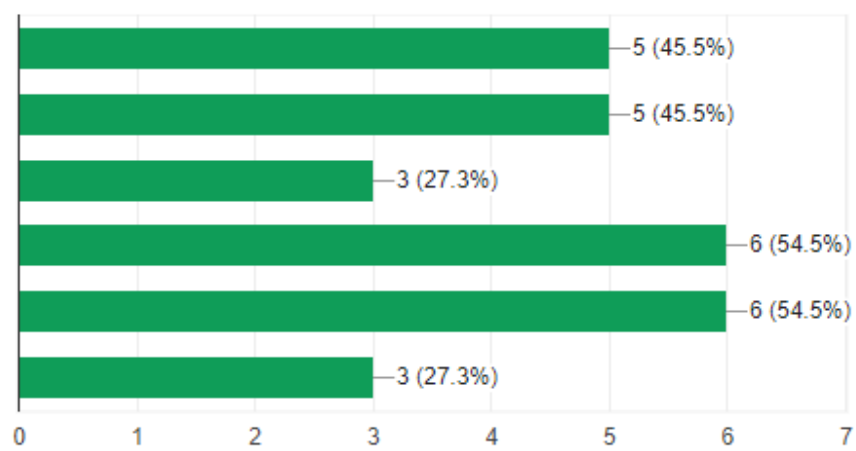
## Would you say you reap benefit from using this tool?

29 responses



**Managers Only\*****How does the task tool influence your management?**

11 responses



## Are there any suggestions on how to improve tasks or support tool? Any new features or requirements would be appreciated?

7 responses

### Integration with work log

The support tool is a good way to log tickets, I have experienced that I have to email the person that is assigned to my ticket to eventually process the ticket, I don't think the tool can be blamed for human error.

Please ignore previous answer. There was not a box applicable to me. Currently the tool is adding to my workload and not assisting me in any way. Due to this tool, emails are being ignored. I work with this tool as little as possible.

Is it possible to be able to create reports on time spent on tasks that we can personally trigger and enable personal evaluations? Yes, we already have the work log platform. But it is more of an input tool and does not give feedback output. So if we can be able to generate a report (bar chart, pie chart, etc), that notifies me of how I distribute my time throughout the week, month. Then I can be able to re-prioritize the tasks.

Can't we make it that when the task is completed it should add to worklog

"Trello-like" project board will make me happy

Actual hours per task must be calculated.  
Employee/worker scoring system for task completion.

## APPENDIX II

---

### SQL queries, calculations and results

```
-- Count the amount of tasks in a certain time frame

select count(t_CreatedStamp) as Month1 from task where t_CreatedStamp < '2017-05-00'
-- select count(t_CreatedStamp) as Month2 from task where t_CreatedStamp < '2017-06-00' and
t_CreatedStamp > '2017-05-00'
-- select count(t_CreatedStamp) as Month3 from task where t_CreatedStamp < '2017-07-00' and
t_CreatedStamp > '2017-06-00'
-- select count(t_CreatedStamp) as Month4 from task where t_CreatedStamp < '2017-08-00' and
t_CreatedStamp > '2017-07-00'
-- select count(t_CreatedStamp) as Month5 from task where t_CreatedStamp < '2017-09-00' and
t_CreatedStamp > '2017-08-00'
-- select count(t_CreatedStamp) as Month6 from task where t_CreatedStamp < '2017-10-00' and
t_CreatedStamp > '2017-09-00'
-- select count(t_CreatedStamp) as Month7 from task where t_CreatedStamp < '2017-11-00' and
t_CreatedStamp > '2017-10-00'
-- select count(t_CreatedStamp) as Month8 from task where t_CreatedStamp < '2017-12-00' and
t_CreatedStamp > '2017-11-00'

-- select count(t_CreatedStamp) as Month9 from task where t_CreatedStamp < '2018-01-00' and
t_CreatedStamp > '2017-12-00'
-- select count(t_CreatedStamp) as Month10 from task where t_CreatedStamp < '2018-02-00' and
t_CreatedStamp > '2018-01-00'
-- select count(t_CreatedStamp) as Month11 from task where t_CreatedStamp < '2018-03-00' and
t_CreatedStamp > '2018-02-00'

-- AND fk_tt_ID IN (2,3,4,5,7)
```

```

-- Total comments
SELECT count(*) as numcommentTotal
FROM task_comment, task
WHERE task_comment.fk_t_ID = task.t_ID
AND task.fk_tt_ID IN (1,2,3,4,5,7) ;
-- Exclude inspection tasks since the comments are used for metadata and not communication
-- 6030 => total including progress, description , and auto-generated comments

--
=====

-- Total task with comments
select count(*) as total from (

SELECT count(*) as numtasks FROM task_comment, task
where task_comment.fk_tct_ID IN (1,3) --general, progress
AND task_comment.fk_t_ID = task.t_ID
AND task.fk_tt_ID IN (1,2,3,4,5,7) --all but inspections
group by task_comment.fk_t_ID
-- Get all the comments and group them per task

) as table1 #count the groups
-- 1361 => Tasks
-- 6030/1361 =
-- 4.4305 comments per task

--
=====

-- Maximum comments per task
select max(maxcom) as total from (

SELECT count(*) as maxcom FROM tasks.task_comment
where fk_tct_ID IN (1,2,3,4,5,7)
group by fk_t_ID

) as table1

-- 44 comments max
--
=====

-- Total comments where the user commented on his own task
SELECT count(*) as numcommentTotal
FROM task_comment, task
WHERE task_comment.fk_t_ID = task.t_ID
AND task.fk_tt_ID IN (1,2,3,4,5,7)
AND task.fk_u_ID = task_comment.fk_u_ID

```

```
-- Closed IT support tickets
SELECT COUNT(task.t_ID) AS Support
FROM task
WHERE fk_tt_ID IN (5)
AND task.t_IsOpen = 0
-- => 181

-- =====
-- Open IT support tickets
SELECT COUNT(task.t_ID) AS Support
FROM task
WHERE fk_tt_ID IN (5) #IT support
AND task.t_IsOpen = 1
-- => 12

-- =====
-- Open support development
SELECT COUNT(task.t_ID) AS Dev
FROM task
WHERE fk_tt_ID IN (2,3,4,7)
AND task.t_IsOpen = 1
-- => 139

-- =====
-- Support task types
SELECT COUNT(task.t_ID) AS Dev
FROM task
WHERE fk_tt_ID IN (2,3,4,7)
AND task.t_IsOpen = 0
-- =>874

-- =====

-- Total support tickets
SELECT COUNT(*) AS TotalSupp FROM tasks.task
WHERE fk_tt_ID IN (2,3,4,7)
-- =>1013

-- => Total support => 189+12+139+874 => 1206
-- => Total open => 12+139 => 151
-- => Completion => 1- 0.125 => 0.874
```

```
-- Count amount of tasks with labels

select count(*) as total from

(
SELECT count(*) as numlab FROM tasks.tasklink_label
group by fk_t_id
) AS table1

-- => 934 have labels

-- =====

-- How many labels are priority

SELECT count(*) as patrk FROM tasklink_label
where fk_tlt_ID = 1

-- => 696 have priority

-- =====

-- How many labels have status
SELECT count(*) as patrk FROM tasklink_label
where fk_tlt_ID = 2
-- => 539

-- =====

-- Count the amount of labels with certain text

SELECT count(*) FROM tasklink_label
WHERE tll_Text = 'Low';
-- WHERE tll_Text = 'Crisis';
-- WHERE tll_Text = 'High';
-- WHERE tll_Text = 'Medium';
-- WHERE tll_Text = 'Busy';
-- WHERE tll_Text = 'Review';
-- WHERE tll_Text = 'Queued';
-- WHERE tll_Text = 'Testing';
-- WHERE tll_Text = 'Future';
-- WHERE tll_Text = 'Await Info';
-- WHERE tll_Text = 'Quick';
```

```
-- Get the count, min, max and sum from all the linked users to tasks
```

```
SELECT SUM(taskcounter) as total -- => 6681
-- SELECT count(taskcounter) as total -- => 2671
-- SELECT min(taskcounter) as total -- => 1
-- SELECT max(taskcounter) as total -- => 71
```

```
FROM
```

```
(
SELECT count(fk_t_ID) as taskcounter, fk_t_ID
FROM TaskLINK_AssignedUser
GROUP BY fk_t_ID
)
```

```
AS table1
```

```
-- Average :
-- => 6681/2671 = 2.501
```

```
-- =====
```

```
-- Get the min, max, sum and count per task type
```

```
SELECT SUM(taskcounter) as total
-- SELECT count(taskcounter) as total
-- SELECT min(taskcounter) as total
-- SELECT max(taskcounter) as total
```

```
FROM
```

```
(
SELECT count(fk_t_ID) as taskcounter, fk_t_ID
FROM TaskLINK_AssignedUser
WHERE fk_tlt_ID = 1
-- WHERE fk_tlt_ID = 2
-- WHERE fk_tlt_ID = 3
```

```
GROUP BY fk_t_ID
```

```
) AS table1
```

```
-- Total amount of tasks linked =>2671
```

```
-- => SUM tlt = 1 => 349
-- => COUNT tlt = 1 => 349
-- => MIN tlt = 1 => 1
-- => MAX tlt = 1 => 1
-- => AVG tlt = 1 => 349/2671 = 0.13066
```

```
-- => SUM tlt = 2 => 1704
-- => COUNT tlt = 2 => 1254
```

```
-- Amount of support ticktes
SELECT COUNT(*) AS TotalSupp FROM tasks.task
WHERE fk_tt_ID IN (2,3,4,5,7)
-- => 1206

-- =====

-- Amount if IT support
SELECT COUNT(*) AS TotalSupp FROM tasks.task
WHERE fk_tt_ID IN (5)
-- => 193

-- =====

-- Inspections and general tasks
SELECT COUNT(*) AS TotalSupp FROM tasks.task
WHERE fk_tt_ID IN (1,6) #inspections and general
-- => 1484

-- =====

-- Amount of users that have used tasks in their lifetime
select count(*) from
(SELECT distinct fk_u_ID FROM tasks.task) as table1
-- => 97

-- =====

-- Users using the system
SELECT count(*) as tot FROM mtb_database.mt_users
where isDeleted = 0
and groupid = 8
-- => 110

-- =====

-- Amount of open tasks
SELECT COUNT(*) AS totalOpen FROM tasks.task
WHERE t_IsOpen = 1
-- =>543 open tasks

-- =====

-- Amount of closed tasks
SELECT COUNT(*) AS totalClose FROM tasks.task
WHERE t_IsOpen = 0

-- => 2147 closed tasks

-- => 1 - 543/(2147+543)= 0.798
```

# APPENDIX III

---

Tasks Manual

# TASKS MANUAL

*Report author: Jaco J. Botma*

## Table of contents

List of figures .....	2
List of tables .....	2
Introduction .....	3
1 Quick Start Example .....	3
2 Web-platform interface .....	6
2.1 Overview .....	6
2.2 Labels and buttons .....	6
2.3 Types of users .....	7
2.4 Types of tasks .....	7
3 Task Configuration .....	8
3.1 Basic Configuration .....	8
3.2 Schedule .....	9
4 Actions .....	11
4.1 Comment .....	12
4.2 Refresh .....	12
4.3 Configuration .....	12
4.4 Assigned Users .....	12
4.5 Complete/Active .....	13
4.6 Task Options .....	14
4.7 Labels .....	14
4.8 Progress .....	14
4.9 Supervisor .....	15
4.10 Informed Users .....	15

## List of figures

<b>Figure 1: Web-platform Main menu</b> .....	3
<b>Figure 2: User Tasks list</b> .....	3
<b>Figure 3: Create Task</b> .....	4
<b>Figure 4 Tasks list</b> .....	4
<b>Figure 5 Assign users</b> .....	4
<b>Figure 6 Add Comment</b> .....	5
<b>Figure 7 Tabs to switch between tasks</b> .....	7
<b>Figure 8: Task basic configuration</b> .....	8
<b>Figure 9: Daily setup</b> .....	9
<b>Figure 10 Weekly Configuration</b> .....	10
<b>Figure 11: Monthly setup dialog</b> .....	10
<b>Figure 12: Monthly weekday calculation</b> .....	11
<b>Figure 13: Monthly amount option</b> .....	11
<b>Figure 14: Sidebar with task actions</b> .....	12
<b>Figure 15: Comment, configuration and refresh buttons</b> .....	12
<b>Figure 16: Assigned User</b> .....	13
<b>Figure 17: Assign user dialog</b> .....	13
<b>Figure 18: Complete/Active button</b> .....	13
<b>Figure 19: Open/Close tab for tasks</b> .....	13
<b>Figure 20: Task Options</b> .....	14
<b>Figure 21: Task Labels</b> .....	14
<b>Figure 22: Label dialog</b> .....	14
<b>Figure 23: Task Progress</b> .....	14
<b>Figure 24: Task Progress Dialog</b> .....	15
<b>Figure 25: Task Supervisor</b> .....	15
<b>Figure 26: Task Supervisor Dialog</b> .....	15

## List of tables

<b>Table 1: Description of task status indicators</b> .....	6
---	---

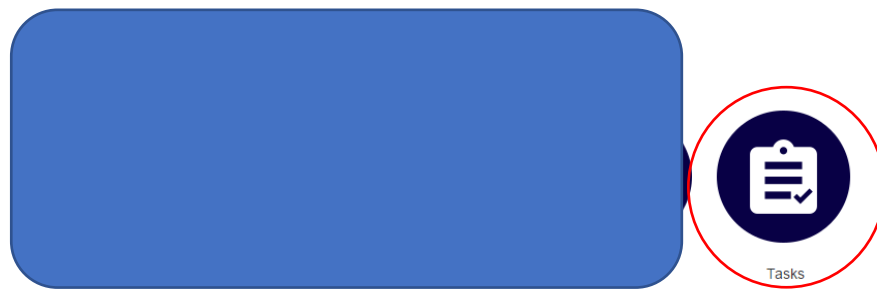
## Introduction

The purpose of this manual is to explain the functionality of the tasks sub-application in the web-platform. Tasks is a multifunctional planning application that allows users to manage issues and objectives, which are related to projects, workflow, and other administrative tasks. Tasks are created with a short title and an optional description. Users are assigned to, or mentioned in, a task to enable its completion. The task itself acts as a forum for the supervisor and assignees to interact and track progress.

### 1 Quick Start Example

Below follows a basic example of how a task can be created. This section does not nearly cover all the functionality of the tool. Please refer to the remainder of the manual for more information.

- In **Figure 1** click the **Tasks** icon to access your **Tasks List**.



**Figure 1: Web-platform Main menu**

- In **Figure 2** click the + icon to add a new task.

Task Name	Assigned Users	Due Date	Progress	Informed Users	Supervisor	Created by	Date Created	Last Changed
Complete Data generation		In 3 days		MC van der Bank		MC van der Bank	Today	Today

**Figure 2: User Tasks list**

- In **Figure 3** Add a short name or title for your task. If an optional description is needed, type this in the description field.
- Click the save button at the top of the screen to finish task creation.
- Additional options such as the due date and recurring options can be added by clicking on the + signs.

Figure 3: Create Task

- Click on the name of the task in the **Task List**. See **Figure 4**.

Task Name	Assigned Users	Due Date	Progress	Informed Users
Complete Data generation <span style="background-color: #4a7ebb; color: white; padding: 2px 5px; border-radius: 3px;"> </span>		In 3 days		MC van der Bank

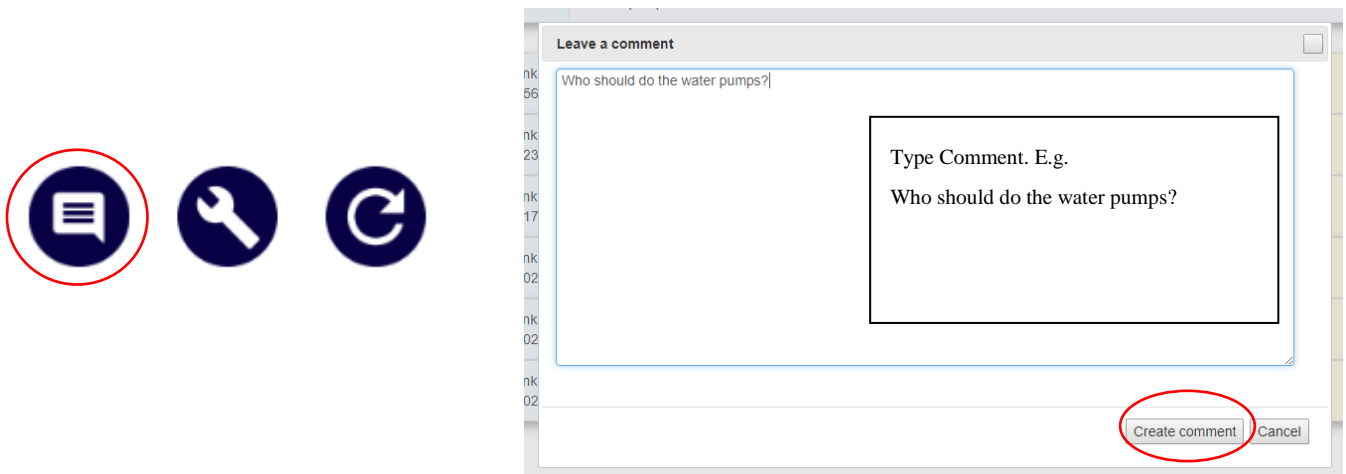
Results per page 20

Figure 4 Tasks list

- Assign users to grant them access and the **Tasks system** will email them a link to the task. See **Figure 5**.

Figure 5 Assign users

- Add a comment to post any questions or confirm procedures. See **Figure 6**. All informed users, assigned users and the supervisor(s) will receive an email containing this comment.



**Figure 6 Add Comment**

## 2 Web-platform interface

### 2.1 Overview

A list of the tasks created, or assigned to a user, can be accessed from the **Tasks list**. In **Figure 2** this is shown. The tasks are ordered by due date closest to the current time. At the sidebar, several filters can be applied to find a relevant task. These filters include:






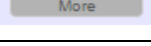
- Due Date
- Labels
- Assigned Users

To apply a filter, click the desired option. To view all the tasks for the user, i.e. remove any filters, click on the filter and deselect the applied filter options.

Tasks are divided into active or completed tasks. These can be viewed by using the tab selector at the top of the list (active tasks are displayed by default). Any completed tasks are considered. However, these tasks are never fully deleted because it provides for traceability and record keeping of any work done.

### 2.2 Labels and buttons

**Table 1: Description of task status indicators**

Item	Description
	Indicates that the assigned users started with the task.
	The task is low priority.
	The task cannot be completed at this stage.
	The task is medium priority.
	The task is high priority.
	The button toggles the viewing of assigned/informed users.

## 2.3 Types of users

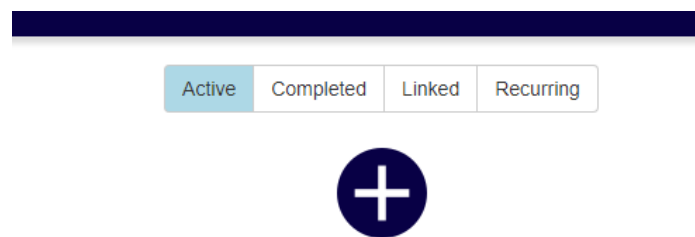
Tasks can have several user types linked to them. These are explained below:

Type	Description
Assigned	Responsible for completing the task.
Informed	Kept up-to-date with anything regarding the task.
Supervisor	Responsible for ensuring the task is completed.

Any user linked to a task will receive email notifications. Any user can unlink himself/herself from the task, if linked incorrectly or by mistake.

## 2.4 Types of tasks

Below in **Figure 7** the types of tasks are listed in the different tabs that is found at the **Tasks list**.



**Figure 7** Tabs to switch between tasks

### 2.4.1 Active

This is the priority type. The **Tasks list** will default to this type. These tasks are in the process of being completed.

### 2.4.2 Completed

After a task is finished, it is marked as finished. It is then stored under the **Completed** tab.

### 2.4.3 Linked

If a user is accidentally unlinked or needs access to a task it previously was linked to, this tab will grant you access.

### 2.4.4 Recurring

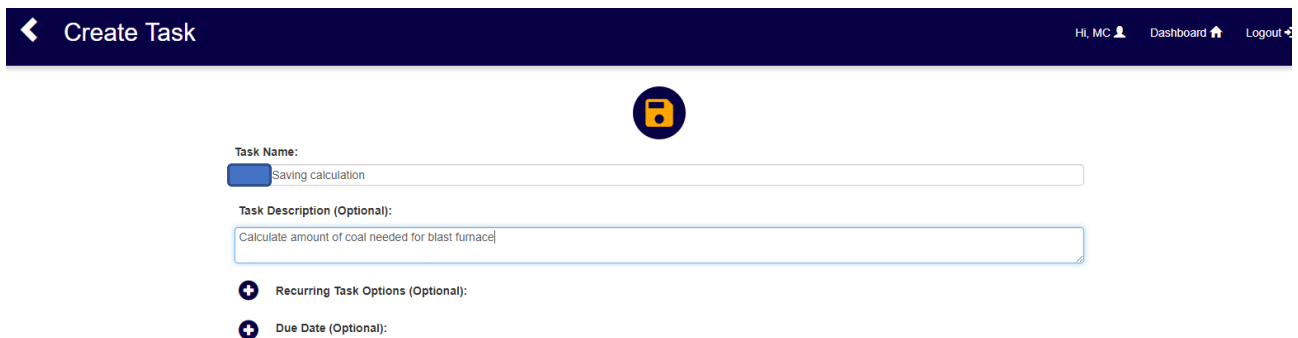
All tasks with recurring schedule options will be found in this tab.

## 3 Task Configuration

### 3.1 Basic Configuration

The task itself has some configuration options. These can be set from the Task Options gear at the sidebar, or when creating the task. Both display the same configuration menu as seen in **Figure 8**.

- The task name is intended to be a short identifier of the task.
- The task description is an optional longer explanation that will display at the top of the task feed.
- To add any additional options, click on the + sign next to the label.
- The task due date is an optional time that can be given to show when the task should have been completed.
- Only the task name is required. Any other fields can be left empty or removed by clicking on the - button on the left.
- To save the current task configuration, click on the save icon at the top of the screen.



The screenshot shows a 'Create Task' interface. At the top left is a back arrow and the text 'Create Task'. At the top right are links for 'Hi, MC', 'Dashboard', and 'Logout'. In the center is a save icon. Below it are three input fields: 'Task Name' containing 'Saving calculation', 'Task Description (Optional)' containing 'Calculate amount of coal needed for blast furnace', and two expandable sections: 'Recurring Task Options (Optional)' and 'Due Date (Optional)'.

**Figure 8: Task basic configuration**

## 3.2 Schedule

To add a schedule to a task, the + button next to the label on the left side must be clicked (see **Figure 9**). This will mark the task as recurring. A recurring task creates a duplicate of itself, but without the comments and schedule settings of the original task. Labels, descriptions and due dates will automatically be calculated.

Once a recurring task is closed, or the schedule options are removed, the task stops duplicating itself.

### 3.2.1 Daily

A task can duplicate itself daily. When a schedule option is changed, a list of next due dates will appear at the right side of the **Recurring Task Options** dialog, as shown in **Figure 9**. The task can recur a set amount of days by changing the number in the input box located below the **Time Frame**.

The create time dictates the time when the task is duplicated. The due date is the time that the task should be done daily. This is shown in hours and minutes that can be adjusted.

The screenshot shows the 'Recurring Task Options (Optional)' dialog. At the top left, there is a '+' button labeled 'Add/Remove Button'. Below it is a toggle for 'Recurring Task Options (Optional)'. The 'Time Frame' is set to 'Daily'. Under 'Recure every', there is an input box with '1' and the text 'day(s)', with a callout 'Recur amount'. The 'Task creation time' is set to '09 : 00'. To the right, 'Difference between recurring and due date:' is shown as 'Hours: 7' and 'Minutes: 20'. Below that is a 'List of next due dates:' with a scrollable list: 'Mon 2017-06-26 16:20', 'Tue 2017-06-27 16:20', 'Wed 2017-06-28 16:20', 'Thu 2017-06-29 16:20', and 'Fri 2017-06-30 16:20'. A callout 'Next Due Dates' points to this list. At the bottom, the 'Due Date (Optional)' is set to '16 : 20'.

**Figure 9: Daily setup**

### 3.2.2 Weekly

A weekly configuration can be created by changing the **Time Frame** to “Weekly”, as shown in **Figure 10**. The next option to select is the day of the week it must be created. From this day, the due date will be calculated. The due date can be removed by clicking the - button.

The due date is calculated from an integer of days after the creation time.

Weekly

Day of the week:  
Monday

Recur every 1 week(s).

Task creation time:  
11 : 00

List of next due dates:  
Tue 2017-06-13 11:00  
Tue 2017-06-20 11:00  
Tue 2017-06-27 11:00  
Tue 2017-07-04 11:00  
Tue 2017-07-11 11:00

Figure 10 Weekly Configuration

### 3.2.3 Monthly

#### 3.2.3.1 All

This option selects a specific day of the month. The task will be duplicated every month on the creation time. The due date will be calculated from the creation time using an integer number of days. If the day does not occur on every day of the month a dialog will appear as shown in the middle section of **Figure 11**.

Information

A Date has been chosen that cannot be consistent throughout the year.

Causes include :

- 31st does not exist for every month.
- February only has 28 days.

Would you like to use:

Same day each month  Remember

Same day each month  
End of each month  
Begin of each month

Apply

Figure 11: Monthly setup dialog

The following options specify when monthly recurrence should occur:

- **Same day each month:** The day will try to stay the same each month. If it does not exist, the closest day in the same month will be selected. The list of next due dates can be used to verify whether the output is as desired. If the 31<sup>st</sup> is selected, all days will land on either 31, 30 or 28/29 (for February).
- **End of each month:** The task will always be created at the end of the month.
- **Beginning of each month:** The task will be created at the beginning of the next month.
- The remember checkbox remembers the selection. If this needs to be undone, refresh the webpage, or remove and add the recurring options again.

#### 3.2.3.2 Weekday

For a task to be created with weekday recurrence settings, the “Weekday” tab needs to be selected as shown in **Figure 12**. This option enables the user to select the first or last weekday of the month for the task to recur on.

The due date is again calculated as an integer number of days from the creation time.

The screenshot shows a form for task creation. At the top, there is a dropdown menu set to 'Monthly'. To its right are three tabs: 'All', 'Weekday' (which is selected and highlighted in blue), and 'Amount'. Below the 'Monthly' dropdown is another dropdown set to 'First', followed by the text 'day of the month that matches:'. Underneath this is a second dropdown menu set to 'Monday'. At the bottom, there is a label 'Recur every' followed by a text input field containing the number '1' and the text 'month(s)'.

**Figure 12: Monthly weekday calculation**

### 3.2.3.3 Amount

The amount tab allows the user to select the first or last number of days from the beginning/end of the month, for recurring task creation. This is shown in **Figure 13**. Selecting the “First” number of days option is essentially the same as using the **All** tab (recommended).

Using the “Last” option allows the user to set the creation time a set number of days from the end of the month. If 5 days are chosen, 5 days will be subtracted from the end of the month to create the next task creation time. Thus, recurring tasks that are created might appear on 25<sup>th</sup>, 26<sup>th</sup> or 23<sup>rd</sup> of the month (depending on the last day of the month).

The screenshot shows a form for task creation. At the top, there is a dropdown menu set to 'Monthly'. To its right are three tabs: 'All', 'Weekday', and 'Amount' (which is selected and highlighted in blue). Below the 'Monthly' dropdown is another dropdown set to 'First', followed by the text 'day number for month:'. To the right of this text is a text input field containing the number '1'. Below this is a label 'Recur every' followed by a text input field containing the number '1' and the text 'month(s)'. To the right of the 'day number for month' field, there is a note: 'Task is created on e.g. 1st of each month.'

**Figure 13: Monthly amount option**

## 4 Actions

All actions are available from the sidebar located on the left, but only once a task has been created and selected for access. To access any task, click on the name of the task in the user task list, which is shown in **Figure 14**.

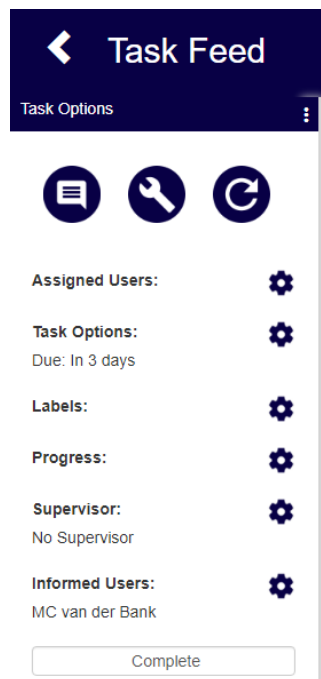


Figure 14: Sidebar with task actions

#### 4.1 Comment

To create a comment, click on the comment button as shown in **Figure 15**. This will open the comment dialog. Once a comment is added by a user, all assigned and mentioned users will receive an email notification.



Figure 15: Comment, configuration and refresh buttons

#### 4.2 Refresh


Clicking on the refresh button, as shown in **Figure 15**, reloads the feed. If a comment on the page is added, whilst another user is viewing, click on this button to view any new items in the feed.

#### 4.3 Configuration

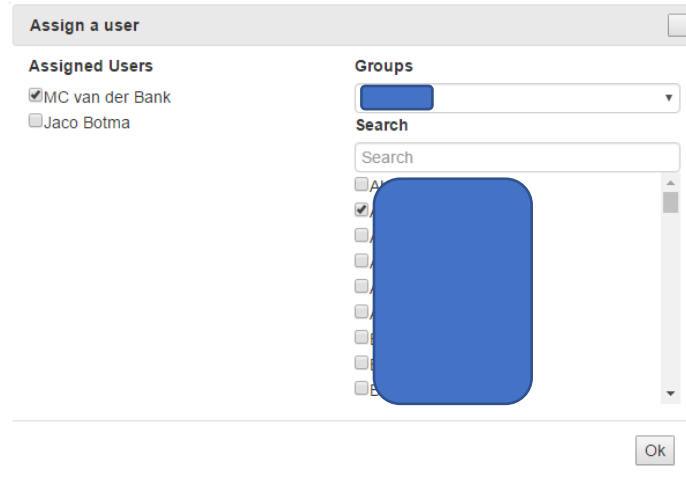
Clicking on the wrench icon located in the middle at **Figure 15**, links you to edit the task properties.

#### 4.4 Assigned Users

To assign a user to a task, click on the gear next to the **Assigned Users** label as in Figure 11. The assign user dialog will then open, as shown in **Figure 16**, and from the **Search** section a list of users can be selected. To un-assign a user, deselect the name from the **Assigned Users** section as shown in **Figure 17** for Jaco Botma. Click on OK to confirm the assignments. Once a user is assigned, he/she will receive an email notification.

**Assigned Users:**   
 MC van der Bank  
 Jaco Botma

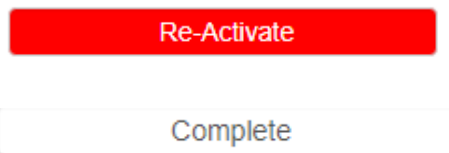
**Figure 16: Assigned User**



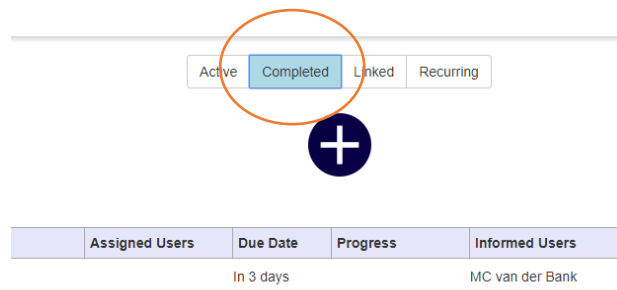
**Figure 17: Assign user dialog**

## 4.5 Complete/Active

A task can be completed and re-activated by using the **Complete** button, as shown in **Figure 18**. An email notification will be sent to all users linked to the task, when the task is changed. When a task is completed, it is considered finished, but it will not be deleted or removed from the records. A completed task can be accessed and re-activated from the closed tab at the user task list as shown in **Figure 19**.



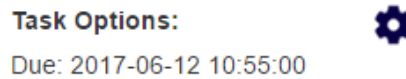
**Figure 18: Complete/Active button**



**Figure 19: Open/Close tab for tasks**

## 4.6 Task Options

To update the due date, task description or edit the schedule options, the **Task Options** gear can be used (shown in **Figure 20**). This area will show if a task is recurring and the due date. More details are given in the *Task Configuration* section.



**Figure 20: Task Options**

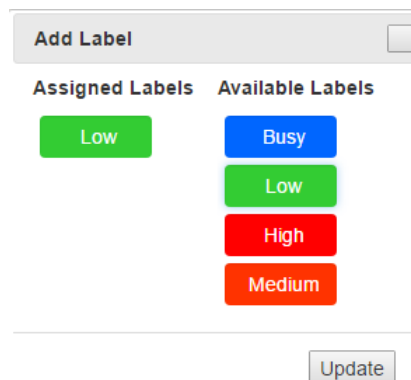
## 4.7 Labels

Labels can be added to a task to indicate additional information. Typically, these can be used to prioritise tasks. To access this feature, click on the gear icon as shown in **Figure 21**.



**Figure 21: Task Labels**

The label dialog will then be opened. To add a label, click on the label itself. This will move the label to the left side, as shown in **Figure 22**. To remove a label, click on the label on the assigned (left) side. After label assignments have been made or removed, click the update button to confirm.



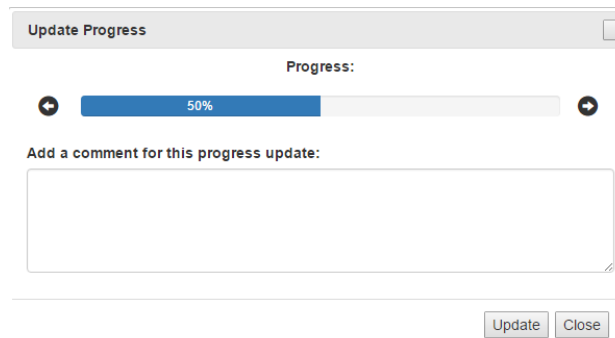
**Figure 22: Label dialog**

## 4.8 Progress

The progress of a task can be updated by clicking on the **Progress** gear, as shown in **Figure 23**. This will open the progress dialog of **Figure 24**. To update the progress of a task, a comment must be made to describe the progress. Click on the progress bar itself to change the percentage value of the progress, or use the arrows on the left and right sides of the progress bar. An email will be sent to all users linked to the task, to notify them of a progress update.



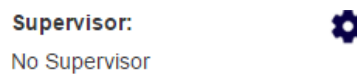
**Figure 23: Task Progress**



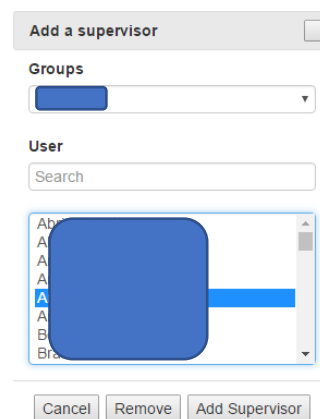
**Figure 24: Task Progress Dialog**

## 4.9 Supervisor

The supervisor of a task can be set by clicking on the gear next to the **Supervisor** label, as shown in **Figure 26**. This will open the supervisor dialog of **Figure 25**. The supervisor can be added and removed from here. Only one supervisor is allowed per task.



**Figure 25: Task Supervisor**



**Figure 26: Task Supervisor Dialog**

## 4.10 Informed Users

Informed users are not responsible for managing or completing the task. This type of user will receive emails on all activity of the task and can view the task. See the *Assigned Users* section for operational instructions.